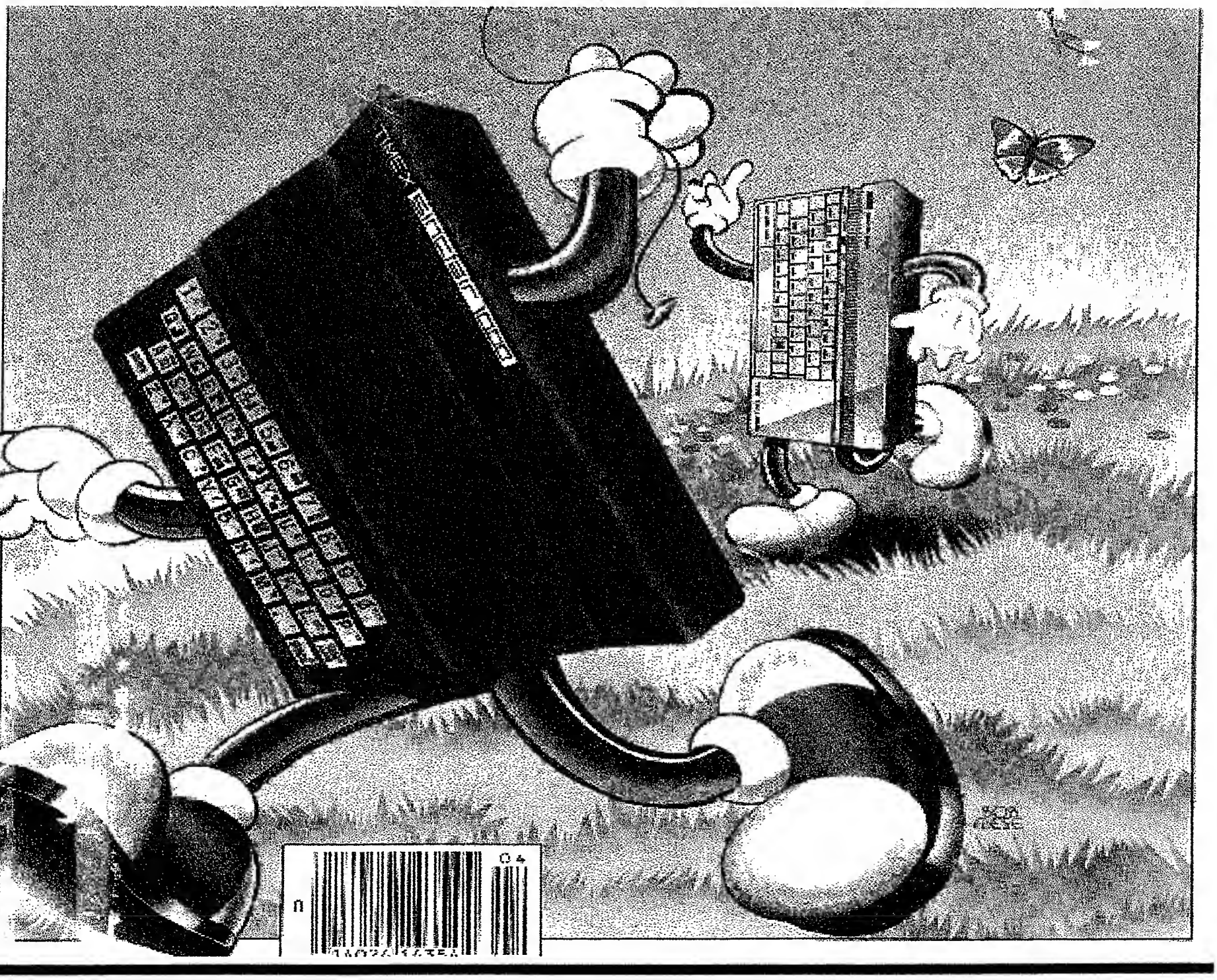
ZXir QLive Alive!

The Timex/Sinclair NorthAmerican User Groups Newsletter

12th Anniversary Final Spring 2003

MEMORY MAP



ZXir QLive Alive! ©

Established 1991 The Timex/Sinclair NorthAmerican User Groups Newsletter

T/SNUG Information

Chairman

Donald S. Lambert
738 Gunnar Ln.
Forsyth, IL 62535
(217) 875-8043
dslambert@email.msn.com

Vice-Chairmen

Tape & JLO PD Library
Luke Perry
3409 NE 62nd Ave. #187
Vancouver, WA 98661

Z88 Library
Dave Bennett (HATSUG)
1275 Timber View Dr.
Mechanicsburg, PA 17055-9146
717 732-4374
dave975@att.net

QL Hacker's Journal
Timothy Swenson
2455 Medallion Dr.
Union City, CA 94587-1914

TS-2068
Rod Humphreys (VSUG)
10984 Collins Pl.
Delta, BC V4C 7E6 Canada
604 583-2819

QL PD Library
John Donaldson (CATUG)
835 Foxwood Cir.
Geneva, IL 60134-1631
630 232-6147
goodolejohn@avenew.com

AERCO & Z80 Emulator
Keith Watson
41634 Amberly Dr.
Mt. Clemens, MI 48038

--==GATOR==--
Bob Swoger (CATUG)
613 Parkside Cir.
Streamwood, IL 60107-1647
630 837-7957
Rswoger@aol.com

ABED KAHALE
432 WEST OAKS TRL
WOODSTOCK GA 30188-7358
AKahale@juno.com

WebPages

www.timexsinclair.org
www.ts2068@yahoogroups.com
www.ql-users@nvg.ntnu.no
www.ql-users@quanta.org.uk
http://geocities.com/nesqlug1/
www.dokos-gr.net/~nesqlug/
http://users.aol.com/clubbbs/tsnug/

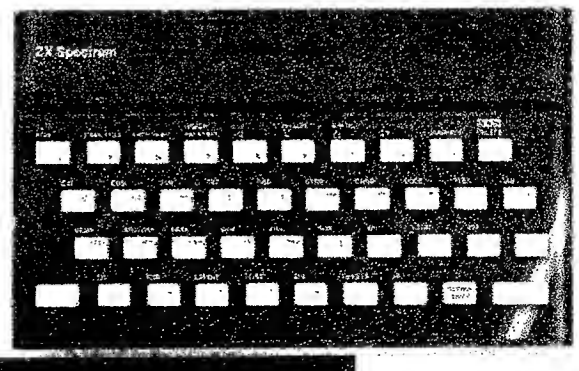
The first personal computer for under $200.

The Sinclair ZX80.
A complete computer—
only $199.95

Introducing the ZX80 kit. $99.95

$99.95
Complete kit

1979



1982

ZQA! Signing Off

T/SNUG To Continue

At 3 AM in the morning ten years ago I was unable to sleep. I was thinking about **SNUG**, the Sinclair NorthAmerican Users Group, and a group that just couldn't get off the ground. Its purpose was to be the preservation of Sinclair hardware and software in the United States of America. The fellows trying to get the group going were young working guys that were trying to keep their families going and could not put enough time into SNUG to keep its members happy. I thought up a new organization to do the things promised by the old group headed up by folks that might have more time to devote to the goal. I wrote a letter that would get a new group started to do what the old group couldn't seem to get going. I would call it **T/SNUG**, the

Timex/Sinclair NorthAmerican User Groups.

It would be represented by the collection of the user groups still in existence at the time and its officers would be from the older crowd that had empty nests at home. It would be the new home for those that would in time lose their present user groups.

Don Lambert was a man who had just retired from Collins Radio and had lost his user group. He had put out a newsletter called the CRAGIST in Cedar Rapids, Iowa and had recently moved to Auburn Indiana. I phoned him and he agreed to become our Chairman. The newsletter name "**ZXir QLive Alive!**" was the creation of **Al Feng** who cleverly brought together the **ZX** of the ZILOG family of machines and the **QL** for the Motorola chip machines. As for the content of the newsletter, I could supply enough to get us started. I had developed the **LogiCall** operating system for the **LarKen** Disk Drive interface which **Larry Kenny** himself told me he wished he had gone with LogiCall instead of the system from Florida. For years I had submitted programs to the T/S rags only to never see them get into print. Now was my chance to get the word out about the power of the LarKen/LogiCall marriage, which allowed the TS-2068 to switch between the Spectrum ROM and the TS-2068 ROM on the fly.

By the way, the final version of LogiCall was released to ender **Frank Davis** of **FWD** Computing in 1995 in time to sell at the Dayton ComputerFest. When I attended the 1996 Dayton ComputerFest, Frank reminded me that it had been one year since Gates released Windows 95. More than 5000 bugs were found in it. Frank then went on to say, "LogiCall has also been out there one year and no one has found any bugs in IT yet." ... And no one ever has! Thanks, Frank! Frank and Carol Davis still do our local "Chicago Color ComputerFest" every year. *Thanks Again, Frank And Carol!*

At 3 AM this morning I am up again, not for the same reason as ten years ago, but rather, to go to the bathroom. I know of no Timex/Sinclair group that still meets. I do know a few guys that still claim to step up to the T/S machines from time to time. My two TS-2068s are presently safely stored away until I can get time to bring them back to life again this fall when I get rid of a couple of high priority honey-do jobs my wife laid on me.

Alas! A next stage for T/SNUG has come to pass. Abed no longer gets input for our ZQA! Newsletter. It is now time to publish and

mail our last ZXir QLive Alive! Newsletter. Abed and I met back in October and again in March.

We discussed how the T/SNUG organization could continue. We decided to hold off on this last newsletter to have time to put the next stage into action. In the mean time, along came Jose Moreno in Miami, Florida requesting that he be allowed to put up a web site called ZQA (he forgot the "!"). We believe he is hoping to keep the e-mail portion of the newsletter alive but to date. I can't figure out how to read the inputs that go to that web site.

Then there is **Callum Davidson** who will simply keep the newsletter on the web site and will send a printed copy only to those who want to pay for the hard copy. I also hope to see him have a LarKen TS-2068 by the end of 2003 to keep his interest up. The Timex stuff is new to him as he is really a Spectrum man. LarKen will let him switch between the two and will enable him to get up to speed on the TS-2068.

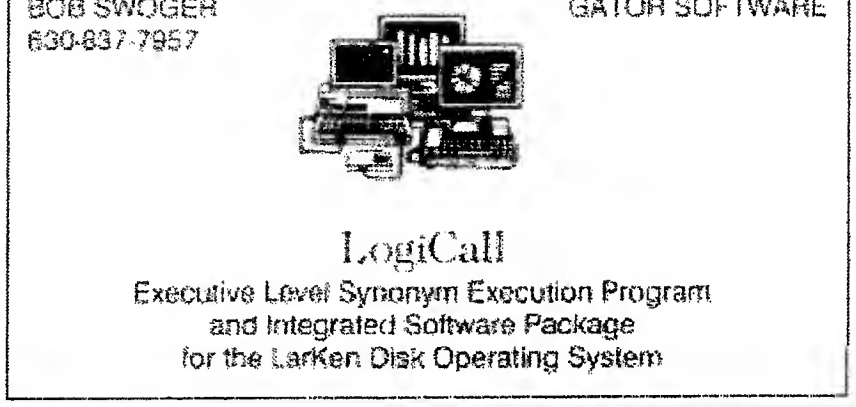
What Abed and I decided to do is to take any inputs we get and put them into a continuously growing newsletter that can be read only on the T/SNUG web site at

http://members.aol.com/clubbbs/tsnug/

No more mailed hard copies, no more dues. We will try to keep you all abreast of what is available in terms of hardware at our remaining hardware storage site in Iowa. Please also remember that you can contribute your T/S related hardware to the site rather than sending it to a landfill. The contact information is in the Unclassified Ads of this last newsletter.

Just get on the Internet and bookmark our web site and check it out from time to time. Now that I, too, am retired, I hope to update our site with changes more often. It really is up to you all to keep things alive, keep e-mailing Abed, and me even just to say hello once in a while. That is what it takes to keep it going. And of course, Abed - thanks again for all your hard work. You have been the greatest asset that T/SNUG has had these past eleven years. And thanks to **J Shepard**, **Jack Boatwright** and all the generous people that have made our hardware availability a reality to this day. We all have been an unusual group of users.

Bob --==GATOR==-- **Swoger,** K9WVY



Back in the early 70's, at my place of work, we (engineering department) were allowed to log on the company's IBM 360 mainframe when it was not in use by the accounting department. I learned BASICA by a visit to IBM in downtown Chicago. Later we had Time-Share and were able to use IBM's downtown computer, logging on using the telephone (50 BAUD) and a printer.

A couple years later, we got CompuServe and a **monitor**, we still had to dial up and enter our code to get online. I never really got to fully put to use the BASICA language because of the limitations on allotted time and the connect charges.

The end of the 1970's, the company got a Commodore 64 for my office. In 1979 I saw an ad in an engineering magazine for the Sinclair ZX-80 for $200, then a ZX-80 in kit form for $99. I got and built that kit pronto and learned the BASIC language.

In 1980, I got the ZX-81 and immediately updated its 1K RAM. There were magazines articles and newsletters devoted to it.

The TS-2068 appeared in 1982, color, 64K RAM, printer, 300 BAUD modem, etc..... I didn't waste any time buying one. Various magazines began to cover all aspects of the TS-2068 and there were plenty of writers and suppliers of add-ons and attachments, and many newsletters. I acquired the LarKen and a disk drive, later on, the LogiCall.

End of 1980's, I joined the Chicago CATUG group. Bob Swoger published a newsletter. Later on he published ZQA!. He was having difficulties in meeting the deadlines of two N/Ls with his full time job, so I, retired by that time, took over the publishing up to this date.

Abed Kahale

The
Timex/Sinclair NorthAmerican User Groups
Newsletter

FROM OUT OF 'THE ASHES' RISES
>>>> *ZXir QLive Alive!* <<<<

Timex/Sinclair NorthAmerican User Groups **Auburn, Indiana**
Volume I, Number 1 Spring 1991

MEMORY MAP

SNUG is DEAD!
Long LIVE T/SNUG!

If you feel as I do, READ ON!
If a doctor saw no life in a body for six months he would presume that body was dead! If our user group doesn't hear from another user group for six months we presume it is dead and we cease to exchange newsletters with it. We have waited long enough for the leaders of SNUG to MOVE but they haven't! Do you all remember "Lead, follow or get out of the way!"?
The folks at the helm presently must not be thought of as BAD, they just don't have the time to devote to this undertaking. Therefore, let us not chase them away as we will need their help, let's just ask them to step aside and let us get moving. As **LARRY KENNY** points out all the time "being negative causes destruction and loss. The SNUG officers are fine men, they are just overloaded.
We have a MAN WHO IS WILLING and HAS the SPUNK to motivate OTHERS who will MOVE. That man is **DON LAMBERT** of AUBURN, INDIANA, formerly of CRAGIST, the newsletter of the CEDAR RAPIDS, IOWA group. T/S user groups, let's give Don his head, as you would let go of the restraints of a horse, and LET HIM TAKE YOU HOME, that is, to a place that gives you a warm feeling!
What I am proposing here is that WE allow Don to be the CHAIRMAN of a NEW organization called T/SNUG, the Timex/Sinclair NorthAmerican 2
VICE-CHAIRMAN from their group to add their names to the list of vice-chairman for the purpose of maintaining activity in T/SNUG. The present conventional officer set-up of SNUG makes no sense! The president has to call meetings and have the other officers present. For a continent wide outfit, HOW? How can you vote them out?
A good number of T/S users are retired and have talent.
If Don took the reigns as Chairman, and other men took a vice- chairmanship for other tasks, Don would not have to wait on others to get an OK to move from a few officers, he could move on his own from what he gets from a pool of about 20. Also, Don has most of the equipment. What say we give him a try? At first he will be the whole thing, but as each member group adds to the list of vice-chairman willing to do some task, this thing can grow.
Don would put out a newsletter at least four times a year.
Don would see to it that software libraries were built up and listed in the newsletter periodically.
Don would build a list of every known T/S user in North America and make it available to all user groups.
The T/SNUG newsletter would always be available on BBS so that all the INFO could be downloaded.
The T/SNUG newsletter would also include printings of vender catalogs FREE to the VENDORS.
CATUG did offer SNUG newsletter help and SNUG did not take it, we offered BBS help and SNUG did not take it. Don will accept help from all of us.
How do we give this a try? Find someone in your group who is retired and willing to send copies of your newsletters to Don in hard copy and 5 1/4" disk in one of the formats he mentions in his letter and see what he does with it. A small check of encouragement would be nice to get this moving.
As for FUNDS? They would be in the future the same as for SNUG and if SNUG has any left on hand now, they could forward those funds to Don eventually. Don must not do this out of his own pocket, for sure, his wife would kill him!

T/SNUG CHAIRMEN

Here is the list of 1991 T/SNUG Chairman and how to contact them. We wish to support at least the following SIGS, ZX80/ZX81, SPECTRUM/TS2068/TC2068 and QL. If you have questions about any of these fine machines contact the Chairman.

POSITION	NAME	PHONE	PRIMARY FUNCTION
Chairman	Don Lambert	219-925-1372	Chief Motivator
Vice-Chairman	Bob Swoger	708-837-7957	Newsletter/BBS Sysop
Vice-Chairman	Al Feng		Nite-Times News/CATUG QL
Vice-Chairman	Rod Gowen		Plotter/CCATS
Vice-Chairman	Bill Harmer		TS Bullitin
Vice-Chairman	Open		QL

Many thanks to **Al Feng** for the newsletter name, **ZX**ir **QL**ive **Alive!** It expresses the wish of many to keep the ZX/TS/QL family of computers going for decades to come.
Bob Swoger, Representative Chicago Area Timex Users Group CATUG

From The Chairman's Desk

Happy 1991! The turn of the decade certainly has heralded many changes. 1991 sure started out with a bang, Desert Storm and all, but one small shocker was a phone call from **Bob Swoger** after he read a letter I wrote to CCATS in OREGON. He asked for a disk with my letter to CCATS on it and here is a portion of that letter:

To: CLACKAMAS COUNTY AREA
T/S USERS GROUP

Dear T/Sers,

Is there a SNUG? That depends upon what you call it. I talked to **Paul Holmgren** and the next issue (is it #3) is about ready to be mailed. I got a firm promise of that before the next meeting of ISTUG but also at the last meeting got assurances it would be mailed within two weeks.

I assure you it does me no good as an editor not knowing when it will get published and mailed so that I will know what material was used in the issue so that I can lay out the next issue. I do have the material for the next few issues and some material of my own to include.

This is not by my design but what has been given to me, I do not have the list of addresses of members or I would have dug into my pocket and gotten something out. For some reason they or Paul wants to retain the control of the printing and mailing. And Paul is 155 miles away and letters go unanswered and telephone calls get expensive. They requested me to be SNUG newsletter editor last November and so far not much has been done. That seems typical of SNUG.

If I had the information to be able to mail out to the SNUG members, something would have been done, at least by last Christmas. Nothing fancy but at least let everyone know that SNUG was still around. Several times I almost gave up but keep thinking that it will get going. I hope that it will before all the T/Sers leave the fold.

About elections, I would accept an office, I don't know the duties of any but I would not turn down any. But I would want to continue the SNUG newsletter if at all possible unless someone with better qualifications comes along. It will be a thankless job I know but if anyone sends me material the newsletter will continue.....

T/S computers, I am still in the beginners stage in many ways. I will admit that since I have attended the ISTUG meetings beginning back in October missed the January meeting, swap meet too) I have learned so much more. Not what I would go to the meetings to find out but what was there to learn and what the others wanted to explain. I still have problems but I am further along than before I moved into ISTUG territory.

I only work with the ZX80, ZX81, T/S1000, T/S1500 and the T/S2068. I do have one T/S2068 with the SPECTRUM ROM but I have not used it. When feel that I have learned the T/S2068 I will be ready for the SPECTRUM. Since the QL and the Z88 do not use the same language I am not interested in them. At present, the only time I would want to have a QL is when I need to copy an EPROM.

I have much hardware for the above computers and lately, March 2nd, I performed a marriage and now have the LarKen/Oliger disk systems [SUPER DOS] running on my working T/S2068 computer. I did not do the complete hardware modification of the LarKen dock board since I did not bend out pin 1 of the 74HCT74 chip on the OLIGER SAFE board and tie it to pin 14 so that I could power up with both interfaces active without the computer freezing up as suggested by Larry Kenny. The only problem is that I cannot LOAD my Oliger version of MSCRIPT V5.5 with the LarKen board enabled. If do and I try to LOAD a file the computer does a NEW! A quirk can live with since I an now aware of it. The reason I did not bend out pin 1 is that the chip is soldered in.

On the ZX81 I am trying to get an AERCO disk interface working with double sided 40 track drives. The used system that have has one single sided 35 track drive and the docs..... the first 5 1/4" drives were single sided 35 track and quite likely single density since the software asks when you FORMAT if it is single or double density. Changing drives requires a new interface EPROM. AERCO still makes the system and supplies the EPROM's. I have the LarKen ZX81 disk system but to SAVE a program that has machine code in it you have to POKE the start address and the length of the code and I do not know how to find that out.

On the T/S2068 I have been transferring my LarKen MSCRIPT files to Oliger by way of a cassette version of MSCRIPT V5.0 which I converted to LOAD LarKen and to SAVE Oliger. That does beat SAVEing to cassette and reLOADing to the other version of MSCRIPT which I did a few times before I got the LarKen/Oliger marriage performed.

Anyone with comments to send in to SNUG please do this, ask for help, got something to sell, want to buy something, got a problem, got a solution to a problem you have had. Anything at all about any of the T/S computers including the clone of the ZX81, the PC8300 (or IQ8300).

If you write and want an answer please use a LSASE so that I do not have to dip into my allowance to reply. My postage bill is high enough and the new rate does eat into my allowance even more. Can't believe how fast the stamps go, I buy them about 40 at a whack and it seems like I have to do it more than once a month. Same with telephone calls, I am willing to talk anytime except that my wife when she is home does like to have me do other things. . . . .

Bob Swoger is the Editor for the Glenside Color Computer Club, the Chicago Area Timex Users Group, and both President and Editor of the Motorola MicroComputer Club in Illinois.

I will need your help, I can't do it alone, but I assure you that when I say that the Timex/Sinclair NorthAmerican Users Group is going to be moving forward in innovative software, user support, and product assistance; you are not hearing the rumblings of a madman nor the visions of a lunatic. You are hearing the convictions of one who knows the binding power of this fellowship, and the collective desire to 'help the person next to me'.

Don Lambert, Chairman
Timex/Sinclair NorthAmerican User Groups

From the Chairman's Desk

It was a wonderful adventure to enter the world of Timex Sinclair computing. I entered it in early 1980's - was it 1982? I owned at one time or another a ZX-80, ZX-81, TS-1000, TS-1500 and a TS-2068. I also had a clone of the ZX81 and later a Z88. And I did have a disk drive system for the TS-1000 but when I acquire a LarKen disk drive system for the TS-2068 I really was into computing. I ended up with several different disk drive systems and finally I was using an Oliger/LarKen system until I left the TS computers in 1999 when we moved from Auburn, Indiana to Forsyth, Illinois. When we moved from Cedar Rapids, Iowa to Auburn I had (my wife's count) 31 boxes of stuff that she declared were my computer system plus backup parts. And when we contemplated the move from Auburn to Forsyth I had to agree to leave the T/S computers and get a pc to get rid of the boxes of stuff. So I shipped a lot of stuff to Jack Boatwright. However, I did keep the Z88 and what I had for the Z88 plus some manuals for the T/S computers.

So from the time I shipped the T/S stuff to Jack till I bought a pc I was in withdrawal without a computer. And that was from the early part of July till September when I had shopped and found the best deal (so I think) at a Sam's Club for a Compaq Presario 5715 computer, which I still use and will as long as I can. The first time I powered up that system I knew I wasn't with a T/S computer. I waited and waited for it to finally be ready to use. And then I had no idea of what to do. My learning curve was flat for a long time. I finally stumbled onto a copy of Smart Computing magazine and I signed up and from that time on I was learning. That was in February 2000.

I haven't forgotten the T/S computers but I only have the Z88 (currently stored) and memories of the grand times I had with them. And all the friends that I made in those years. And all the trips I made to go to the T/S computer meetings and especially the trip to Washington, DC and to Milwaukee and the many trips to Dayton. Ah, those were the days!

The only thing that I want is to be able to interface the Z88 to the pc. I have the software and the hardware but attempts to upload to the pc fail. Or maybe it doesn't fail and I don't know where to find where the file went. A 13 gig hard drive is quite large and it could be there and I don't know where it is or the file's name. That would make it easier for me to manage notes and stuff since my handwriting is degrading from bad to worse. And along that line I even bought a Sharp YO-520 organizer thinking that if I could manage to upload memos from it I could learn how to do it with the Z88 BUT that too didn't work.

I also have a Laser PC-3 which is a poor man's version of the Z88 that is supposed to be up loadable to a pc but I somehow lost the disk of software for the pc (it was a 5.25 floppy) and my pc uses 3.5 floppies) so that would have been another problem. However I do have PC Tools by Central Point PC Tools, which was the company that wrote the software for the PC-3 back in the first part of the 1980's. Currently I am studying DOS for the PC to learn how to load the proper software program into the pc. If someone could give me any details of how to do the upload from the Z88 to a pc in simple English I would greatly appreciated it.

I have a working disk drive interface for the Z88, which would make it more useful to use if I could upload to the pc.

Health wise I am healthy now but there was a time when I wasn't. Back in 2002 in October on a Sunday I felt something wasn't right and had my oldest daughter take me to the hospital. That was the 27th. On the 30th they performed a 5-way bypass on my heart and an aorta repair. They had to stop my heart to be able to work on it (the heart refused to cooperate) so a 3.5-hour operation ended up being a 7+-hour operation. I was a grouchy bear for quite a while since I couldn't do anything nor did I feel like it either. And I am pretty much back to where I was before the operation although at my age (77) I don't have the stamina that I had before. I did not have a heart attack but probably would have had soon if I hadn't went to the hospital for a checkup on why I felt like I did.

Donald S. Lambert

They never heard of the Sinclairs !!

Input/Output

by Abed Kahale

US QL SHOW 2003

Quanta and NESQLUG are pleased to announce the US QL show to be held Saturday 17 May 2003 from 9 AM to 5 PM at the Econo Lodge at 370 Highland St., West Haven, Connecticut 06516-3522. West Haven is on the coast adjacent to New Haven. The special rate at the Econo-Lodge is $59 (including tax!) per room per night for 1 to 4 persons if you make reservations before 17 April. Call 203 934-6611, email: econolodge@comcast.net or mail. Please mention "Albert rate" and include your credit card number. Continental breakfast (coffee and pastry) is included.

New Haven Tweed (HVN) is the closest airport, but the closest international airport is 50 miles away - Bradley International in Hartford, CT. The New York airports JFK and La Guardia are a little over one hour away. Newark Airport in New Jersey is not much further but requires a ride through New York City. NESQLUG will endeavor to provide rides for those arriving by air. Please contact Bill Cable, email cable@cyberportal.net if you need a need or can help out with a ride.

The Econo Lodge is 2 miles from the beach. From the north, take I-95 exit 42, take right turn to Route 162 East, hotel is a half mile on the left. Several restaurants and a shopping mall are nearby. Those who arrive by 6 PM Friday may optionally meet in the parking lot to eat together in a recommended restaurant.

Nearby New Haven is the home of Yale University and contains several museums and other tourist attractions. Many other attractions are along the Connecticut coast, plus there is good and cheap public transportation to New York City. Ladies will meet at 10 AM to make plans with Dorothy Boehm to see nearby sights. Contact Al Boehm, tel: 256 859-8051 or email albertboehm@juno.com for further information.

Albert R Boehm
albertboehm@juno.com

I just received a letter from the sister-in-law of Frank Mills. She told me that Frank passed last July of a stroke. His wife Jo moved to Goreville, IL to live with the sister-in-law, Molly Zalar - Molly said we can send cards to Josephine Mills, 55 Lake Shore Dr. S., Goreville IL 62939 (1-618-995-2399)

Molly said that Jo read our Christmas Letter and responded to tell me that Jo's health is poor, she has breast & spine cancer and is taking radiation.

I will always remember Frank, he was a gentle guy with the ability to laugh at the antics of us younger guys back there in the 80's. He used that TS2068 mainly to do his banking using the LarKen RAMdisk with no disk drive. I had to write a special program for it.

Frank was 90.5 years of age, a retiree of Motorola. A grand old age I'll miss him.

--==GATOR==-- **Robert E. Swoger** - K9WVY

From: Les Cottrell
Thu, 06 Feb 2003 18:39:45 -0500
I have posted an eBay listing for a 2068 "superchip" at http://cgi.ebay.com/ws/eBayISAPI.dll?ViewItem&item=3400580738&category=4193

Les Cottrell
lcottrelljr@cfl.rr.com

Hi all,

In response to Johnny Red's question about 5 1/4" 80 track drives, I have two 5 1/4" drives that are DSQD, or quad density drives. Radio Shack sold a box of 10 5 1/4 diskettes for a while that were specifically made for these type of drives. I found that I could normally use the DSDD diskettes just as well and they didn't cost as much either. I believe these DSQD drives were found mostly on CPM type computers. I used them with my Timex AERCO FD68 drive system which allows 35, 40, 76 and 80 track drives in 3", 3.5", 5 1/4" and 8" sizes (but not in that order). All drives on the AERCO FD68 can be single or double sided except for the 8" drive which can only be single sided. I've used 3" single sided (and an unusual 3" double sided version), 3.5" single/double sided, and 5 1/4" double sided drives with my FD68 system. With a little jumper modification to old 1.2 meg HP 5 1/4 drives, it is possible for these drives to work with the AERCO FD68 system as 80 track 800 Kbytes drives.

The AERCO FD68 system can also boot with a look-a-like CPM system which is called RPM. You do get 80 columns on the screen but it is pretty slow to display on the Timex screen. It is very much better to use a terminal display and turn off the Timex screen. Then the display really flies.

Keith Watson
keithwatson@netzero.net

TS2068 Enthusiast

Greetings to you all and thank you for allowing me to join your group! I first owned a TS2068 in 1983 as a young'n. My dad would not buy me the ever-popular console gaming systems and told me that the only video games I was allowed to play were those I could program in BASIC on the TS2068. Coincidentally, I became adept in BASIC programming.

I gave that system to a church in 1990, while I was enlisted in the Coast Guard. I recently regained interest in the computer while surfing the web.
I bought my current one on e-bay, and I've made it my mission to maximize my TS2068 over the next year or so. The hardest things to acquire will likely be a FDD system (preferably 3.5 disks), an adapter for a modern printer, and (if possible) a hard disk drive-- yes, I've my work cut out for me...

I look forward to knowing more about each of you. If you want to know more about me, here is my rather pathetic website:

http://www.geocities.com/grey_six/jay_and_sherri.ht

ml?1035781498145

John Simpson

jegs_and_sherri@msn.com

Hi Abed

It's several weeks ago that I have received the Autumn 2002 issue of ZQA! I was really sad to read that you will stop with your nice magazine.
I got in touch with several friendly people through ZQA! Though there wasn't too much about my favorite computer ZX81/TS1000, I enjoyed to read about the activities in the USA. Especially I likes to read the articles about Z88. Be sure I will miss ZQA!

We do also have the problem getting articles for our ZX-TEAM-MAGAZINe from most of our members, but fortunately there are some, who support us on a regular base. And you may have seen in the last issues, I could take some information from the internet too.

So we will keep on editing our magazine. Unfortunately I do not have the time to translate to English and I didn't find someone to do this job, so everyone who doesn't understand the German language, will be unable to participate.

Please tell me if you still want to receive the ZX-TEAM-MAGAZINe in future. I would be glad to send it to you! Good by(t)e

Peter Liebert-Adelt

http://www.zx81.de

Hello Everyone....

I am still interested in working with HiSoft Pascal programs but I am out of ideas on what kind of programs you need. Send me some ideas and I will try to develop them.

David Solly

k_david_solly@hotmail.com

Hello Abed,

Paul Holmgren and Frank Davis have given an official okay for the

QL Emulators CD

published by RWAP Software of West Yorkshire, UK to be distributed in North America with the QL ROM images under certain conditions and in return for a small royalty. The ROMs, software, hardware and name of Sinclair Computing and all trademarks still remain the protected intellectual and copyrighted property of Paul Holmgren and Frank Davis, till the end of 2007. This should make the use of QL emulators for those who have not owned a legal QL easier to set up as long as all conditions are met.

Frank Davis

FWD Computing

fdavis@iquest.net

www.fwdcomputing.bizland.com/

Abed --

I just learned from Jack Boatwright that you are terminating publication of ZQA. I am certainly sad to hear this news.

Help

I would like to ask if you would please include a request for information in the final issue. Some time ago Jack sent me a LarKen 1000 floppy disk drive controller for the TS1000/ZX81. The device appears to be faulty.

I am in need of the ROM code (in *any* format), schematic, documentation, and *any other information or knowledge* regarding this device. Thanks for your help.

Glen Goodwin

acme@ao.net

Hello Abed,

I would be interested in articles on ZX81 emulators and WRX16 Hi-Res Video, especially!
My address is:

Mark Anderson

319 South Grove

Mora, MN 55051

pfourier@mninter.net

Hi Don!

Long time, no see. I hope to find you well and still focus on Sinclair/Timex products. If all goes well (and it does), ZX91 and ANDRE*** should be back this coming fall with all the goodies as before plus release of new programs of 3 kinds Free-Ware, Shareware and Pay$-Ware. But my final decision is not made up yet on how to make them available.

Hope to ear from you and will give you more details. As you see I am now playing on internet. Keep the good work, Best regards.

Andre Baune

ANDRE***

zx81ab@progression.net

2003-04-03

Abed,

There is a set of freeware QL emulators for DOS and Windows and Linux downloadable from: www.inter.nl.net/hcc/A.Jaw.Venema

These work fine except reading program files is a little klunky since you first have to get them into QL format. But there is a separate program QLtools that does it for you. Smoother is the low cost emulator for the MAC on:

http://users.infoconex.com/daniele/q-emulator.html

which also has low cost ($22.50) and a more versatile ($40) emulator for Windows. There is also the high end emulator, QPC2v3 (EUR 99 which is about $110 US) available from:

www.qbranch.demon.co.uk or http://qpc.j-m-s.com

Also available, but more expensive, is the advanced QL hardware the Q-40 and Q-60 with things like Ethernet cards, stereo speakers, 128 MB ram, and 32 bit graphics. Info available from: info@q40.de

There is also a ton of freeware programs available. See the above site for links. God bless,

Al Boehm

TS-2068 Emulator

I want emulator for TS2068, not the Spectrum. I had a Spectrum deal on my TS system but rarely used it. The

Spectrum was no way as good a machine as the TS2068. My concern is to save my own programs written on TS2068...to be honest. Still, I will check them out if I ever fire my TS2068s up again. Both my Dell and iMac have crashed this week and Dell has been sent back to manufacturer. Thank God for revivable Macs.

Promise still holds --$100 for knowledge of TS2068 emulator for Apple or Wintel. and with bad experiences with Dell, I would prefer TS2068 to Mac emulator. I would simply transfer my progs over to Mac and still give Callum my TS stuff, books and all the rest, as we talked about for Spring Cleaning plans.

Oh, yeah, two disk systems I have are Zebra drive and LarKen drive. Both were working last time I tried them.

Joan Kealy
hjkealy@rionet.org

You can load real software from tape into Warajevo (maybe others, but I haven't checked) and save it onto your system. There are several utilities that are available to let you do this if the emulator you choose doesn't support it, but most do. PlayTZX and Taper are both the most highly regarded tools to convert software from tape to emulator files (.tap and .tzx are exact images of a real tape, and can be used to make tapes from if you need to do so later) Warajevo v2.51 Available from: ftp://ftp.worldofspectrum.org/pub/sinclair/emulators/pc/dos/warsp251.zip>ftp://ftp.worldofspectrum.org/pub/sinclair/emulators/pc/dos/warsp251.zip

M.E.S.S. Available from: <ftp://ftp.worldofspectrum.org/pub/sinclair/emulators/pc/windows/mess0612b.zip>ftp://ftp.worldofspectrum.org/pub/sinclair/emulators/pc/windows/mess0612b.zip This can be used on Apple or PC platforms - the above link is for the Windows version, with the Macintosh one being available from: http://mess.emuverse.com/

Each of the emulators I mentioned are TS-2068 emulators that run on either the PC or Mac - Warajevo is by far the most accurate, but the most slightly more difficult to configure. MESS is very straightforward and works for the Mac and PC platforms. There are literally 100s of ZX Spectrum emulators, but very few TS-2068 ones, and these are they. The only other I'm aware of is MultiMachine (PC Only) which is available from WoS as well, but you'll probably find Warajevo is better. All of them include ZX Spectrum emulation, and usually other models as well; the TC-2048, etc. Virtually all emulators other than these do every variation of ZX Spectrum, but not Timex. Those listed also do both Timex and ZX Spectrum 'natively' without re-emulation, independently of each other. Hope this helps!

Callum Davidson
callumdavidson@hotmail.com

I've owned various Sinclair products since about 1983, and have fairly recently had my interest re-kindled in them and the early home-computer industry. I'm originally from the UK and moved to Tennessee about 5/6 years ago so, naturally enough, I started educating myself about the US market as it compared to the UK one I was familiar with. Alongside this, I began to build up a collection of things I used to own, and things I always wanted to own but could never afford, for their "nostalgia" value more than anything else. As far as the UK goes, I know the market, industry and products as well as (probably better by now) almost anyone else in the US - half because I was there, and half because of the huge amount of time I've spent in the past few years "reminding" myself of what it was all about through research, etc.

The flip-side being that, by comparison, I know virtually nothing about the Timex machines and the US industry from the same era, and it's proven to be surprisingly difficult to track this information down online. With the exception of a few (admittedly, very good) Timex-specific sites, there doesn't seem to be much information or material readily/publicly available. This surprises me since there seemed to be a lot of interest in the machines at the time, and they are radically different to their UK equivalents. There are countless Spectrum emulators, software titles, user guides, manuals, magazine archives available, but hardly anything about the Timex machines, and this bothers me.

So, I joined the ZQA Mailing List shortly after it was set up so I could improve my knowledge of these machines, etc. and to learn from the people that were actively involved with them while I was over there, blissfully unaware, fiddling about with the Spectrum. I can contrast what they know to what I do, and vice-versa, which helps us all out because we each learn something new that can be passed along for the benefit of everybody. I find this very rewarding, and it gives me a "warm-and-fuzzy" to know that there are still people interested in something so old and "useless" by comparison to the PCs of today. That we've come this far in +/- 20 years astounds me, and I like to contrast the type of things we're taking for granted now with how unimaginable they were then, which puts everything into perspective to me in a fairly humbling way.

Nowadays, I write code for a living and can do things at the push of a button that hadn't even been visualized 5 years ago, let alone 10 or 20. I have a 2-year-old Son, and the things he'll be doing will have their roots in what I'm building today. Seeing the distance we've already covered, and knowing that the "next generation" platforms I'm building now will amount to no more to him in 20 years than a ZX-80 does to users today is quite inspiring, and I can't wait to see what he has at his disposal when the time comes.

Harriet, clearly to me, has a long-standing interest and enthusiasm for the Timex machines, and wants to make sure they're not relegated to the "oh yeah, them too" category, and that anyone that expresses an interest or curiosity in them has a place to go for information. I suggested a few emulators a while ago that might be of use to her, and she emailed me a couple of days ago to say she may need some guidance when she starts working with them. I'm no expert on the emulator front, but I'm certainly happy to lend a fresh set of eyes to any problems she might have with them, and can help figure out a solution (if there is one) if possible. I believe she has been unwell recently, and wants to transfer some software she wrote way back before clearing her storage facility of all

the old items she has boxed up and getting dusty, which is why she wanted to investigate the feasibility of using an emulator at all.

I get emails all the time from people asking questions or looking for some piece of information they know they've seen but can't find anymore. I've accumulated quite a lot of material that I can check for references (and I enjoy this, because it provides me with a "excuse" to fiddle with old machines!) and can put people in touch with others that I think might be able to help when I can't. I'm forever mailing books, lists, tapes, etc. to people because I have spares and they've asked for them. I just give away most things because I know those who are asking are genuinely interested in making use of whatever it is they need, and they're not just transient "collectors" building their next eBay inventory - I don't deal with those people on any terms, and can usually spot them coming a mile away.

I've been trying to locate a lot of old magazines, catalogs, newsletters, etc. recently to build a proper list of references and would like to scan some of these in and share them (copyright issues notwithstanding) in an online library of sorts at some point. Also, there seems to be a lot of people looking to find the inevitable missing issue(s) of their otherwise complete collections, and they're always the ones somebody else has a mysterious extra copy of! There's volumes and volumes of otherwise "lost" information lying at-the-back-of-the-garage all over the country that would be really interesting to read and refer to in future, so I'd like to concentrate on building some sort of a central archive for these things. Long-term vision, maybe, but it would be very worthwhile, in my opinion.

So, that's my 2c on the whole Timex/Sinclair-thing. Bit of a rant, I know, but it's one of those things I get a lot of enjoyment from, and I like to think there's still a place for altruism :)

Callum Davidson
callumdavidson@hotmail.com

WN Richardson & Co
6 Ravensmead, Chalfont-St-Peter
Gerrards Cross
Bucks, SL9 0NB, UK
wnr@compuserve.com

Bill Richardson will have usual Z88, QL units, spares etc. and is looking for your redundant Spectrums and parts, particularly Interface Ones and connectors, please. He will also bring RGB, monitors and dot matrix printers if asked for at £15 each, and other bits and pieces.
Traders ads Qbranch:
www.qbranch.demon.co.uk/showflyer.htm>

Tony Firshman
http://www.firshman.demon.co.uk

Hi Abed,

The only thing I have to offer for the last issue of ZQA is that I will always remember my TS-1000 fondly and continue to use it today, but not very often. It has been a good friend. But the people I have met through the users groups are just wonderful, fantastic, kind people, and I will always remember them most fondly since they are very special friends!!! Take care, Joe.

Joseph Rampolla
jprampolla@blazenet.net

Twister Board

In ts2068@yahoogroups.com, Hector Jorge Picone <dunkel_piky@y...> wrote:
Hi, Anybody can help me?
I'm trying to make a twister board for my TS2068.

Greetings, Just a bit of Twister history.

I designed the original "Super Twister" board for Zebra back in the day. Most of the electronics design was done by members of a user group in Long Island, I collected all the circuits and put them together on one board. I don't remember all of the details but it had memory decoding for a replacement ROM, a voltage regulator, an external reset button, some sort of video port and some other circuit, the purpose of which I do not remember. Sorry but I don't have the documentation anymore.

The board was designed to fit into a VHS video case with cutouts for the connectors. As I recall, the 2068 was the same color as a silver '82 Honda Civic. I made a few prototype cases and painted them with a can of DupliColor paint. They sure looked slick but were too labor intensive for production.

I also wrote an extended BASIC that was called WindowPrint 32/64 which Zebra sold. You can find a copy on the Warajevo web site somewhere. The 64 column software was advertised to work with the OS-64 cartridge Zebra developed, actually it was a 64 column driver which could replace the OS-64 cartridge. Naturally, the one line BASIC command to kick the 2068 into 64 column mode was never published in the manual.

There are several PCB prototype houses now that do nice work, some of them even have free downloadable design software. It might be worth looking into if you want to create a new twister board. Regards,

John B.
"j1b3200" j1b3200@yahoo.com

Alvin,

I gave my Dell laptop to my grandson for med school and bought an Apple iBook nearly a month ago. He knew MS computers and I prefer Macs so any emulator must be able to transfer to my Mac computers. Sorry, but Callum Davidson was first so I am spending the $100 to send my TS-2068 stuff to him. He had offered to pay postage and shipping, but I said the reward would be used to cover it. I just this week sent him my tiny TS-2040 printer and last week a box load of the thermal paper. I shall try to figure out if I can make you cassette copies of my music, but I may have to depend on Callum to reproduce those tapes--he is welcome to circulate them. I confess I did some dern good translations for the TS2068 with chording along with lead plus the lyrics going on the screen as the music played.

Astounding that that little computer was so fine. When I consider that I bought each of my TS-2068s for roughly $100 apiece and have spent over $1300 each for three Macs and a Dell...not counting software and all the extras we are constantly buying. In many ways I enjoyed my TS2068s more.

Harriet J. Kealy
hjkealy@rionet.org

Can NASQLUG Take Up The Slack?

by Al Boehm

It is with considerable sorrow that I heard that ZQA would put out its last issue. All of us that enjoy using the T/S computers will be a little diminished. I still remember how my ZX-81 solved a tough math problem I had worked on for years and how my kids grew up playing games on our TS-2068.

Now most of my computing is on the QL. Well, actually most of it is on QL emulators - the QXL, QLAYW, Q-emuLator, and QPC2. Yes, I have Windows and MAC and use them when they are the best tools for the job at hand. However, I find for development work the QL far exceeds anything available on the PC or MAC.

I'll give you one example. I write games for the youth group at church. Pretty easy and fun to do, with SuperBasic. The kids can run them on the QLAYW freeware emulator.

In recent years, the New England Sinclair QL User Group has revised its operations to meet the changing times. One change is that we went to Virtual Meetings via email. This allows members who could never attend a meeting in New England to have a say in things. Some of the debates have been quite vigorous.

Actually, most members live outside New England. So we changed our name to North American Sinclair QL User Group - **NASQLUG**.

NASQLUG sponsors an annual QL show. Many of the European vendors and developers attend and demo the latest hardware and programs. This year the show is in West Haven, CT on 17 May. Last year it was near Washington, DC.

The NASQLUG Journal is published quarterly. It has 6 to 8 pages of technical and news articles. The Journal is a semi-ezine. That is, about half the members elect to get a paper copy but many only obtain the articles via the Internet. However, all can obtain the articles from our web page, which means no more keying in those long programs from the paper copy.

NASQLUG is a sub-group of Quanta, the large QL group centered in England. This affiliation has several advantages. The two most important are: 1. Quanta helps out financially with the annual US QL shows. 2. All the technical articles published in the NASQLUG Journal are also submitted to the Quanta editor who publishes them in the Quanta magazine if they appear to have broader interest. Nearly all of the Journal articles have been reprinted in Quanta. It is not required that a NASQLUG member belong also to Quanta although Quanta membership is encouraged.

QL development is supported by NASQLUG with seed money grants and prizes. For example, NASQLUG provided Simon Goodwin some seed money to develop a MIDI (Musical Interface) capability for the QL. He was able to devise a Keyword to send MIDI compatible signals out of the QL NET port. All of the numerous MIDI music available on the Internet can now be played from a QL hooked to a synth or MIDI keyboard.

There is a NASQLUG web site. In fact, several of them! The old site:

http://geocities.com/nesqlug1/

has last year's Journal articles available for viewing and downloading. It has links to individual member sites such Herb Schaaf's Math site. The link to my MIDI site doesn't work since Juno stopped providing web space. However, our new Webmaster Phoebus Dokos is loading everything on our new site:

www.dokos-gr.net/~nesqlug/

the idea is for everyone who wants to can have a link to their own site. Others can send material to Phoebus to upload to the master site.

NASQLUG is a non-profit organization plus we have learned to produce and mail the Journal at a very low price. The basic annual dues are $12 to include a paper copy of the Journal or $5 for Internet access only. To provide an incentive to authors and officer volunteers, dues are reduced by $5 for anyone that promises to write at least one article per year or volunteers and is elected as an officer. The reduced rates are thus: $7 for paper copy and FREE for Internet access only.

All members of a household become members if one is a member. Children have played important roles even as officers. We had an article on Turtle Graphics to let elementary school age children do simple programs. Also once we found a wife that stayed in her motel room during one of the QL shows. Since then we have always provided activities for family members that are only marginally interested in computers.

Our officers currently include: Director, Treasurer, Show committee, and Webmaster. But several important jobs are empty: Secretary to those without web access, Program Librarian, and Editor (Articles are now published as is without checking for errors!). Moreover, if anyone sees a need, for example a Grant Officer to vitalize and monitor grants, then they can volunteer and be voted into office. Indeed, I have always desired that most members become an officer (A worker Bee! You only get out as much as you put in.).

If you want to join NASQLUG, contact the Treasurer,

Kevin O'Leary

Box 172
Meriden, NH 03770.
olearyk@cyberportal.net
For more information contact Director

Al Boehm
2501 Ermine Drive
Huntsville, AL 35810
albertboehm@juno.com

NESQLUG

To: <goodolejohn@LIGHTFIRST.com
John, (Donaldson)
Sorry to hear about the nursing home. Bobbie and you too will be in my prayers.
The QL emulators that run on a PC are getting so good that rarely do I use my SuperGoldCard QL. The article below tells about them. With them on my PC, I feel I have the best of both worlds.
NESQLUG is doing pretty well mostly because we have change our mode of operation. We are spread out all over the country. A few of us do see each other at the QL show (usually in the Spring) and at a overnight Fall meeting at Bill Cable's in New Hampshire. But most of our meetings are Virtual Meetings done by email every other month.
If you would like to join NESQLUG, the fee is $5 for email only and $12 for paper copy of the NESQLUG Journal. Send to the treasurer: Kevin O'Leary

Five QL Emulators On My PC; Which is Best?

by Al Boehm

Why would anyone have five QL emulators on one PC? Well first there was the QXL. At the time it was the only emulator on a PC. I am calling it two emulators since I run it with two operating systems. Next I bought Q-emuLator for my MAC laptop. So when the PC version came out, I bought it since it was similar to what I was used to. Later I bought QPC2 because it could do some things that Q-emuLator could not. Finally I downloaded the free QLAYW because I wanted to use it on the PCs at the church hall to let kids play QL games I had wrote.
When I realized I had all these emulators on one machine, it looked like a good opportunity to present an unbiased comparison.

My desktop PC has a CYRIX 5x86 CPU running at 120MHtz with 24M ram and a 2.5 Gbyte hard drive plus a SyQuest Ezflyer 230MB removable drive. It has an early Windows 95 OS. A hot machine in 1996, now it is considered out of date. But a major reason I keep it, is that is has ISA slots that I can put the QXL into. Most of the new machines no longer have an ISA slot.

I have bad eyes. So I use a 17 inch monitor which provides a picture 12.4 inches wide. By comparison, the original QL monitor had a picture just over 9 inches wide. Plus some of my preferences are based on the fact that I can not see small fonts. This is a personal thing, but I can't help but believe that there are aging QLers out there with similar problems. In any case, I have the screen set for 800 by 600 pixels which affects some of the screen sizes mentioned below.

The **QXL**

My QXL II has 8M ram. It runs quite well in DOS mode. It can also run from a Windows 95 DOS window but more slowly and with some limitations and glitches. The QXL has it's own CPU. Thus internal processing times are independent of the host PC. However the host is used for I/O and in particular the screen display. So some timings are affected by the host.

The QXL does have NET ports which net well with standard QL's, SGC's, etc. However, the QXL NET uses different circuitry so that the current DIY MIDI will not work on it.

A major drawback of the QXL is that it can not access PC files on the hard disk. Instead, special QXL.WIN files are formatted on the hard disk which contain QL type files with their special dataspace headers on executable files. Many of the QL CD ROMs being sold contain a QXL.WIN file so that the QXL can access them via the PC's CD ROM reader. A major plus for the QXL is that it reads QL and PC floppy disks automatically. One bad quirk is that it will not detect a second PC disk after a PC disk has been read; a QL disk needs to be read in-between to set it straight.

Actually three QXL OS are installed on my hard disk SMSQ, SMSQ/E v2.83, and SMSQ/E v2.98. I rarely use the SMSQ OS which came free with the QXL. Instead I use SMSQ/E which has additional features such as the PE and changing screen sizes on the fly. However, there was a major change between SMSQ/E v2.83 and v2.98; namely enhanced color drivers. This is why I call the QXL two different emulators for comparison below.

For me and my QXL setup, the enhanced color drivers were a step backwards. Through some type of incompatibility, the enhanced colors do not work with my QXL. They do work OK with QPC2 on the same machine!?!? Also the enhanced color drivers slow down screen writing quite a bit as detailed below. Worst for me and my bad eyes, the enhanced color drivers provide the same small font for QL, EGA, VGA, or SVGA screen options.
The 512 QL window is only 7.6 inches wide for v2.98

but 10 inches for v2.83.. Thus I have QXL 2.83 and QXL 2.98 as separate emulators in the comparisons below.

Software Emulators In General

Q-emuLator, QPC2, and QLAYW all act like Windows programs. You start them like a Windows program and can switch back and forth between them and other Window programs merely by clicking the mouse. Q-emuLator and QPC2 allow text saved in the scrap area to be transferred to the paste area of Windows and vice versa. With a little finagling, they can access the Windows files on the hard disk or CD. Each can read QXL.WIN files but only QPC2 can write to a QXL.WIN file.

They all do a good job of running QL programs although they each have some limitations. They each can BEEP through the PC's speaker and can access the parallel and serial ports, but none have a NET port. They each have a provision for using the PC mouse input. Their speed depends on the host PC. On the newer PC models, it is blazing fast, but each has a way of slowing things down for playing games.

Q-emuLator v 2.1

Daniele Terdina wrote Q-emuLator first as a MAC based emulator. Then a Windows 95 based emulator. It requires a file containing a QL ROM. I use a JSU ROM by permission of Frank Davis who owns the North America copyright. Amstrad which owns the copyright for the rest of the world has given permission for non-commercial use. It also comes with an older Minerva ROM which I understand is freeware. I use TK2 since Tony Tebby has given permission for use in emulators. Q-emuLator can not run SMSQ/E although it handles the PE extensions without any trouble.

Q-emuLator has only an original QL 512 by 256 window. However, it can display the window in two different sizes. One is 7.7 inches wide on my 12 inch wide screen and the other is 10 inches. This additional width is of great importance to me and my eyes. Daniele advises me this would be even wider if I had my PC set to 1024 by 768.

One of the unique features of Q-emuLator is that it you can change win assignments on the fly. For example, WIN1 can be changed from meaning the directory C;\QL\docs to meaning floppy drive a: QPC, QLAYW, and the QXL have WIN definitions assigned at startup and can not be changed except by rebooting.

Executable files run OK when copied from QL disks into standard window directories. Daniele has some chicanery here that is beyond me but greatly appreciated. The Q-emuLator Manual is remarkably readable.

Q-emuLator was $60 but recently has been reduced to $40. Also Daniele plans a $22.50 version that only emulates an unexpanded QL. Q-emuLator can be obtained from: http://users.infoconex.com/daniele/q-emulator.html.

QPC2 v3.03 [VER$(1) says 2d99]

Marcel Kilgus wrote QPC2 to run smoothly in the Windows environment. It is a top notch program as the timings below show. Nevertheless it is not without its shortcomings.

The most important limitation to me is in the screens. QL, EGA, VGA, and SVGA are all the same small font size as on the QXL 2.98. This could be a plus for someone who likes to have several small windows open at the same time. For example, running QPC2 with a small QL window to provide answers to put into a small PC window along side it.

QPC2 runs in SMSQ/E only. Most of the time I admire the extra capability of SMSQ/E. But every now and then I wish I could shut it off when there is an incompatibility with older programs. For example, until recently the Turbo compiler would not run on SMSQ/E and even now I can't get programs Turbo compiled on SMSQ/E to run on other systems. Perhaps George Gwilt has already got this sorted out, and I just haven't gotten the word. In any case, I still, on occasion, run into incompatibilities.

Note, I am not complaining about the additional capabilities of SMSQ/E which I use all the time; it's just that I wish QPC2 had a capability to run QDOS when desired.

I have only had QPC2 v3.03 since the Oxon Hill Show. So I haven't had time to delve into all its added attractions. QPC2 v3.03 is available from J-M-S for EUR 99.90 or QBranch for 65 pounds.

QLAYW 0.9c

Jan Venema wrote QLAYW as a freeware Windows program. There are also QLAY versions for MS-DOS and Linux. Since QLAYW is freeware, I can send it to others who can run my QL programs on their PCs.

For me, the best attribute of QLAYW is that the QL 512 screen fits all the way across my monitor - 12.3 inches. Only with the Aurora on my other system am I able to get such a large easy to see screen. Moreover, the screen can be adjusted to any smaller size by using the mouse. There is no noticeable aliasing of pixels and fonts - this is really quite remarkable. On the other hand, only the original 512 by 256 screen is supported.

QLAYW boots up noticeably quicker than the other two software emulators. So if I just want a QL to make a quick calculation, it's my choice.

One bug is that the QL pointer occasionally gets separated from the PC mouse pointer which is disconcerting. Since I rarely use the pointer (I do use the PE!) on the QL, this is not a problem for me.

I found the manuals hard to follow mainly because they bounced back and forth between the DOS, Linux, and Windows versions. The Windows quickstart text once I found it, was easy to follow.

The poorest part of QLAYW is the way it handles files. Files must first be read in by an off-line program - QLtools or QLayt - to make a special directory (qlay.dir) which keeps track of header info such as dataspace for executable programs. This is not much different than the QXL.WIN files except that the QXL can handle floppies directly. I had so much trouble understanding how to do this that I wrote my own reminder which I include:

How to make files available to QLAYW

Make sure QLAYW is not running.
Choose a directory to put the QLAYW win files in.
Copy Qltools.exe into this target directory.
Open DOS from Windows and navigate to target directory.
Have files on QL floppy disk in drive a:
Type: QLtools a: -q
Start QLAYW selecting the target directory for a win drive.

The target directory will contain all the files that were on the disk. The first time QLtools is used it makes the file called qlay.dir which has header info for the files. Further uses of QLtools adds files to the directory and updates qlay.dir Additional file manipulation can be done with qltools.exe and qlayt.exe. I haven't had much experience with QLAYW. Most programs seem to work OK. QD with the PE works OK. Quill works OK. However, Xchange would not run - I guess it had to do with the RAMdisk that Xchange uses.
QLAYW and the other versions (including source!) are available on:

http://www.inter.nl.net/hcc/A.Jaw.Venema

Math Speed

I use a simple double loop to time 1 million math operations. First I run it with just a simple replacement to get the overhead time:

```
100 b=23:c=49: REMark arbitrary constants
110 StartTime=DATE
120 FOR I=1 TO 1000
130 FOR J=1 TO 1000
140 a=b
150 END FOR J
160 END FOR I
170 OverHead=DATE-StartTime
```

Next I change line 140 to a=b+c to get addition time after subtracting OverHead. Then change line 140 to a=b*c to get multiplication time after subtracting OverHead. This timer was run with SBasic/SuperBasic and as a Turbo compiled program. Times are in seconds.

SuperBasic/SBasic

	OverHead	Additions	Multiplication
QXL v2.83	14	10	14
QXL v298	13	8	12
Q-emuLator	1142	256	295
QPC2	62	34	33
QLAYW	1200	250	307
		Turbo Compiled	
QXL v2.83	10	6	12
QXL v298	10	6	11
Q-emuLator	75	46	125
QPC2	52	24	28
QLAYW	70	39	143

The most notable difference is how fast SBasic is compared to SuperBasic.

The QXL times are not a true comparison since they depend solely on the QXL CPU not the host PC. Apparently Marcel has done some fine optimization to get those fast number crunching times.

Scroll Speed

To time screen operations, I used the original QL 512 by 256 MODE 4 screen on all emulators. I scrolled in Quill through the MIDIPlayer manual, a 5382 word, 539 line document. Also I scrolled in ED the MIDIPlayer_bas program of 797 lines. The results in seconds were:

	Quill 539 lines	ED 797 lines
QXL 2.83	17	38
QXL 2.98	43	95
Q-emuLator	28	37
QPC2	37	63
QLAYW	76	145

I think you can see why I prefer the earlier v2.83 on the QXL. I guess the enhanced color drivers are what slow QPC2 down with respect to Q-emuLator. However, it was in MODE 4 and used none of the fancy colors.

Conclusions

All of the emulators are really quite usable. They all have small quirks and limitations. All required about an hour of configuring, tinkering, and reading the manual to get them running. Although QLAYW is not the equal of the other emulators in many categories, it has a smooth feel to it and is easy on my eyes. Moreover, for my intended use for it running QL games on other PCs - it is the best. I can configure it for a particular game and put it on a disk so that a Windows user thinks it is a native PC game.

I still favor the QXL and use it the most. However, QXLs are becoming rare, and so are the PCs with ISA slots to hold them. I am very thankful to the software developers that brought about these emulators to make my fun more interesting and my work easier. I hope this article aids the competitive spirit to make the emulators even better. God bless,

Albert R Boehm
albertboehm@juno.co

Timex Emulation

by Callum Davidson

The number of ZX Spectrum emulators available must be running into the hundreds, with dozens more being produced each year. Given the huge popularity of the Sinclair machines throughout Europe, this isn't really a surprise. What is more surprising, however, is that there are very few emulators available that support the TS-2068.

This may be due in part to the relative rarity of the original machines, a limited awareness of their unique features by members of the wider Sinclair community, and a comparative shortage of original documentation being available online that can be used as a complete reference while developing the emulator itself. While this is certainly improving, the Timex machines are nowhere near as well publicized as their Sinclair counterparts.

Fortunately, the quality of what *is* available is very high indeed, and emulator authors appear to be increasingly interested in the Timex machines and how they differ from their Sinclair counterparts. This is good news, and I hope we'll see more, and better, emulators released in the next few months that provide support for the TS-2068.

Until then, those of us that can't wait should find one of the following emulators suitable for our "everyday" needs:

PC (MS-DOS) : Warajevo v2.51

Warajevo is, by a clear margin, the most complete and accurate TS-2068 emulator available. Unfortunately, it's also the oldest, and can be a little difficult to use unless you do so regularly enough to become familiar with it.

Each key feature of the TS-2068 is supported directly, with several "plug-ins" being available to supplement the core functionality. Note, however, that third-party peripheral emulation is a little lacking. Emulation of Disk Interfaces is limited, as is Printer support.

In reality, these issues don't necessarily represent a major problem for most users; original cassette software can be transferred directly from tape, through the soundcard fitted to your PC, and loaded into the emulator. From there, it can be saved in a variety of emulator-neutral file formats and used directly. Transferring software from disk is more difficult, and loading from cartridge is not possible. Each of these problems can be worked around with a little time and creative thinking, so should not be seen as a major barrier.

Warajevo is designed to run under MS-DOS, but can be used with Windows. I have had no difficulty in running it under Windows XP Professional, for example, without the need to adjust any settings directly, or forcing the program to run in compatibility mode.

The documentation included with the emulator is very extensive, and covers every feature in great detail. In addition, a lot of technical information is provided that describes the way in which the emulation is achieved. The .dck file format was developed to allow "virtual" cartridges to be loaded – this format is fully documented, and is the standard for all TS/TC-2068 emulators with cartridge support.

PC (Windows)

If you are prepared to trade features for convenience, or are interested in more than just the Timex machines, I have found M.E.S.S (Multi Emulator Super System) provides a good compromise.

The TS-2068 is emulated well, with support for standard tape and cartridge files (in .dck format). The source code for M.E.S.S is freely available, and can be modified to suit your own requirements if needed.

Since M.E.S.S is an emulation framework, literally hundreds of machines can be emulated by simply "plugging-in" the appropriate ROM image. No ROMs are included with the distribution, although these are readily available (see below). Any other supported systems can be emulated by simply placing the ROM file in the 'Bios' folder and choosing that system from the application window.

Unix / Linux

There are no dedicated emulators available for Unix / Linux users that specifically support the TS-2068, but all is not lost. The latest version of Fuse, v0.6, emulates the TC-2048 and TC-2068, as introduced to European markets by Timex Portugal. These machines, while not identical to the TS-2068, have some similarities and may work well for you.

As a momentary diversion, the TC-2048 is similar to the original ZX Spectrum, but includes the additional screen modes found in the TS-2068, and a slightly enhanced version of Sinclair BASIC. Note that there is only one Joystick port on the TC-2048, and several internal differences in the two designs that are more significant. There is no cartridge port, for example, so .dck files cannot be used. I have found this model to be slightly more compatible with the ZX

Spectrum than the TS-2068, providing access to many of the software titles released for the Sinclair machine; games, utilities, etc. In addition, many titles that depend on the additional features of the Timex machines (high resolution screen modes in particular) run without any problems, and BASIC is very well supported.

The TC-2068 is, at first glance, the same as a TS-2068. As a general rule, the internal differences between the two designs have no obvious effect on the emulator, and I have had no problems with any software incompatibilities, etc. so far.

One thing to consider, however, is that there will be timing differences between the TS and TC models that may need to be accommodated if you are developing your own software that depends on particular events occurring at fixed intervals. Similarly, simply because the differences in design do not appear to outwardly affect the emulator does not mean that they are of no importance! However, TC-2068 support will suffice until true TS-2068 support is included. Now, back to Fuse....

One of the most appealing features of Fuse is that the full source code is available. Therefore, if the emulator doesn't work the way you want it to, or is missing a feature you feel you need, you can make changes to the source, recompile it, and make your own custom version to suit your particular needs. Not an insignificant task, admittedly, but certainly worth noting.

Apple Macintosh (OSX)

Fuse has been ported to the Macintosh platform (OSX only) and is feature-identical to the Unix / Linux version. Similarly, a version of M.E.S.S is also available, which includes the features described earlier.

Fuse is certainly the most accurate of the options available, but provides only "TC" emulation, not "TS", albeit in addition to a range of ZX Spectrum models. The same caveats still apply, but should not prevent experimentation!

Summary

Each of these emulators offer a similar set of features; which one you choose will largely depend on your current Operating System, and personal preferences.

The relatively limited number of emulators available on each platform, and the fact that they are all available at no cost, means that you can download several and evaluate their features in context of your particular requirements.

It is worth bearing in mind that, thanks to the huge number of ZX Spectrum emulators available, and the relative ease with which support for the European Timex machines can be added to them by authors, that having one of these available as an additional option can prove worthwhile. Normally, unless I know that a particular piece of software makes use of Timex-only features, I'll try running it using Spectaculator v5.0 – this is a superb product that stands apart from other emulators, in both features and accuracy, and I encourage any PC users to download a copy and try it for themselves. TS-2068 support *may* be added in future, but is not currently available. Regardless, I have found that many programs will work unmodified, with the obvious exceptions.

If you have programs that make extensive use of Timex-specific features, try them on the TC-2048 or TC-2068 if these options are available in your emulator. On the PC, I find vbSpec v1.80 provides good TC-2048 support, and most Timex software seems to work well. Only in rare circumstances, or if I need to work with cartridge-sourced software (.dck files), do I find I need to use Warajevo at all.

Macintosh and Unix / Linux users are somewhat more limited in their options. In essence, if neither Fuse nor M.E.S.S work to your liking, tracking down an alternative that does is likely to be quite difficult. Hopefully, this will be a rare occurrence, and there will be more widespread support for these systems in future.

Conclusions

The Timex emulation scene is relatively healthy, with several emulators being available on each of the most common platforms that will meet most users' needs, most of the time. Fortunately, most emulator authors like a challenge, and seem quite prepared to consider requests for additional functionality. They will likely add TS-2068 emulation to their products if people seem to want it, so get typing!

Equally, if you find that none of the current offerings work as you need them to, a short email to their developers should bring results that ultimately are of benefit to us all.

References

The emulators listed in this article, plus thousands of original Timex/Sinclair programs are freely available from a variety of sources. Most emulators have full documentation included, and a small selection of software to get you started.

The following sites provide a wealth of information, utilities and software that you may find useful:

Emulators, Software, Documentation, Utilities, etc. http://www.worldofspectrum.org/
http://www.worldofspectrum.org/emulators.html
http://www.worldofspectrum.org/archive.html
http://www.worldofspectrum.org/warajevo/index.html

Reference information, file format specifications, etc:
http://www.sinclairfaq.com/

Emulators:
http://www.spectaculator.com/

http://www.srcf.ucam.org/~pak21/spectrum/fuse.html
http://www.mess.org/
http://www.muhi.org/vbspec/

ROM Images:

http://www.srcf.ucam.org/~pak21/spectrum/roms.html

General Timex Reference:

http://www.outlawnet.com/~jboatno4/welcome.htm
http://www.timexsinclair.org/2068/
http://timex.johnnyred.org/

Finally, anyone wishing to contact me is welcome to do so.

My email address is:

callumdavidson@hotmail.com

C Programming and the Timex/Sinclair 2068

by Alvin Albrecht

It took nearly 30 years but the de facto standard programming language of the software industry has finally made it to the humble Timex/Sinclair. It has been a long time since I have been this excited about anything in the T/S world and actually being able to write programs in C for the TS-2068 over the past few months has brought a warm grin to my face. What makes it possible is the *Z88DK Small C Cross Compiler* ("http://z88dk. sourceforge.net"). Z88DK supports a nearly ANSI-C implementation of the language and comes with many of the standard C libraries built-in for a variety of Z80 machines, including the Spectrum and ZX81. The level of support for each platform varies depending on the interest paid by the platform's users. The Spectrum has seen some of the most support and boasts a fairly complete set of standard libraries.

This is not the first time a popular high-level language has been made available for Sinclair machines. David Solly has previously written articles about programming the TS-2068 using HiSoft Pascal. Indeed, HiSoft also sold a C compiler for the Spectrum (HiSoft C). The problem with these implementations is that both the compiler and source code had to fit into 48K of memory. This led to compromises, the most important one being that the compiler only implemented subsets of the actual language. Having to split up large programs into many small pieces that could be compiled in memory and then hand linking all the pieces together also made compiling any serious program a serious hassle. Z88DK is a cross compiler, meaning it actually runs on a third computer (like a PC) and generates programs for the TS-2068. Since the source code is entered on the PC and compiled there, Z88DK has the necessary available memory to be as fully featured as desired and can compile source programs of any size with little hassle. There is still the matter of transferring the finished product to the TS-2068 itself but that is also straightforward with the right tools.

For the uninitiated, C was originally developed by Dennis Ritchie in the early '70s to write the first Unix operating system on the DEC PDP-11. It was intentionally made a small language and, as far as high-level languages go, it is as low-level as they get (Forth would be the only example of a lower level language that I can think of). This last point does not mean C is a hard language to pick up like assembly language, but that it deals with the kinds of objects that the bare CPU deals with: bits, characters, numbers and addresses. For these reasons C inherits many of the advantages of both high level and low level languages: it is easy to learn, C programs can be written quickly and the compiled result is fast and compact. The fact that C is such a small language manifests itself when it is realized that nothing much useful can be done with it without additional libraries. Libraries are vendor or user supplied collections of subroutines. Without these libraries there would be no way to print a character, read the keyboard or draw a line. The C language provides the structure of the program and the libraries provide the essence.

C took off like wildfire in the mid '70s and like anything that grows out of control, pretty soon many variations of C not quite compatible with each other arose. The solution was a standardization that occurred in 1983 with the adoption of ANSI-C. ANSI-C formalized various language constructs and clarified many (but not all) implementation issues for compiler writers. Also part of ANSI-C is a standardized collection of libraries (the "standard i/o library") that grants C programs basic i/o functionality (character output, character input, etc.) and access to useful utility subroutines (such as conversion from strings to integers, allocating memory, etc.). With compilers written to adhere to the standard made available for all computer platforms, significant C programs could be written with cross platform compatibility.

Since ANSI-C was introduced the language has evolved into C++, now the standard among desktop and larger computer software. C++ adds language constructs that support object-oriented, generic and functional programming styles. C, like Basic, is an imperative or procedural language where programs are written as step

by step instructions. Unlike C, C++ is a large language and takes many months (some would say years!) to master. Because of its size and complexity it is an inappropriate language to program with on small computer systems.

Z88DK supports a nearly complete subset of ANSI-C and supplies most of the standard library functions expected by C programmers. I have felt little restriction in writing C programs with Z88DK, retaining the typical style I use when writing programs for modern systems. The small nature of the underlying machine means you do have to be more aware of memory constraints, speed issues (i.e. efficient algorithms are important!) and data type sizes (ints are only 16 bits). It is also helpful to know a little about how the C is compiled into assembly language in order to write efficient C on the T/S. Z88DK also supports embedding Z80 assembler into C programs.

What are the advantages of using C on the TS-2068? The biggest advantage is speed. There is simply no way to write anything faster without resorting to assembly language. With C you will also have access to many structured programming constructs that are lacking in Sinclair Basic, like while loops and functions with local variables, that make the programming experience easier and more expressive. Z88DK's libraries also supply many functions that have no Basic equivalent. For example there are functions for printing formatted output, reading formatted input, searching strings and classifying characters as numbers / letters / punctuation. Non-standard library additions allow drawing of lines, filling screen areas and manipulating graphic sprites.

C is not only a significant step up for Basic programmers, it is also a significant tool for assembly language programmers. Complete programs can be written very quickly with excellent performance. If the performance needs to be improved, critical sections of the code can be rewritten seamlessly in assembler. Typically 90% of the time a program executes 10% of the code. Only having to rewrite that 10% of code in assembler to squeeze out the last vestiges of speed can reduce development time from months to weeks. There is also another secret: most of the library routines are written directly in assembler, meaning the C program is mainly used to hold together calls to already optimized assembly language subroutines. This is why C programs on the TS-2068 can be so quick.

I have played cheerleader long enough now; how does one get started? The best reference for learning C is **The C Programming Language** by Kernighan and Ritchie. I have the second edition, which is refreshingly compact having just 200 pages full of examples explaining all the features of C. Most computer books I own near 1000 pages of text and have little content; the reverse is true of this one. The authors do stress that the book is not meant as an introduction to programming but as a presentation of the C language itself. You do not need to buy a book to learn C -- your local library will have many books to choose from and the internet is full of introductory tutorials on C. If ZQA manages to survive past Abed's service to the community I would likely volunteer to write an introduction to C in this newsletter.

With some sort of C reference in hand, the next step is to get hold of the Z88DK compiler. Z88DK is distributed in both binary and source code form. Binary distributions include an EXE for Win32 console with other distributions for the Amiga, Linux and Unix systems. Visit "http://sourceforge.net/project/showfiles.php?group_id=29 17" to reach the Z88DK download page. Grab the latest version (1.5) by clicking on the appropriate link. Windows users should grab the Win32 version. The following discussion pertains to installation for Windows users.

With the zip file download, extract all the files in the archive to some directory (I suggest "C:\z88dk"). Create a file "init.bat" containing the following text and save it in "C:\z88dk":

```
SET PATH=c:\z88dk\bin\;%PATH%
SET Z80_OZFILES=c:\z88dk\LIB\
SET ZCCCFG=c:\z88dk\LIB\CONFIG\
```

This batch file will set up the environment for Z88DK when you are ready to compile your C programs.

I also suggest creating a "work" directory in "C:\z88dk" to contain your C programs. You can write your C programs using any text editor such as Notepad in Windows. When you are ready to compile, open up a dosbox and change directories to "C:\z88dk". Run "init.bat" to set up the environment, then change to your work directory where your C program is saved. Compile it with the following:

```
zcc +zx -vn myprog.c -o myprog.bin -lndos -lm -lsplib2
```

Where:

+zx - tells z88dk to use the Spectrum's standard libraries

-vn - tells z88dk we are not interested in verbose messages

myprog.c is the name of the C program(s) to compile. This can be a list of files if your program is split across many files.

-o myprog.bin - specifies that the output machine code program should be named "myprog.bin"

-lndos - links in the dummy i/o stubs for the Spectrum (essential). The Spectrum does not support all the i/o facilities expected by C.

-lm - links in the floating point math library (optional)

-splib2 - links in the Sprite Pack library (optional)

Z88DK will create a machine code program assembled to run from address 32768 by default. Z88DK will automatically attach machine code subroutines from the libraries listed in the compile command as they are needed. The math library ("-lm") is needed if your program manipulates floating point numbers. Integer operations do not require the math library to be linked. The Sprite Pack library ("-lsplib2") is a non-standard library that contains a collection of machine code routines that I have written over the years. It does not come with Z88DK -- you must download it from my web page at
"http://justme895.tripod.com/main.htm".

The Sprite Pack zip file contains the complete source of the library and can be customized for the various video modes of the TS-2068, among other things, by editing the "SPconfig.def" file prior to compilation. The header file "spritepack.h" contains the function prototypes for all the functions available from Sprite Pack. Compilation of the library creates the "splib2.lib" file. The zip file comes with a precompiled version of the library using the default settings in "SPconfig.def". Copy "splib2.lib" to "C:\z88dk\lib\clibs" and "spritepack.h" to "C:\z88dk\include" to get access to the library. Sprite Pack contains many library functions that support sprites, dynamic memory allocation, IM2 interrupts, keyboard / joystick / mouse scanning and a host of other functionality which makes writing exciting programs for the TS-2068 much easier. The best way to learn how to use it is to go through the examples in the "examples" directory of Sprite Pack. The web page will also have a tutorial in the near future.

We have assumed that you made no mistakes with your C program in order to get the resulting binary file. If you did make some mistakes you will learn about them through error messages. Z88DK compiles C programs in a two step process: first the C is translated into assembler and then the assembler is compiled into machine code. If you've made a mistake you are likely to hear about it twice. To save some headache and deal with the C errors only, compile your programs with "zcc -a -vn myprog.c" at first. This command asks to translate the C to assembler and then stop there. Then only the C errors will be reported.

Once you have a binary file ("myprog.bin") the next step is to run it. The most convenient means to test your programs is to run it on an emulator on your development machine. I typically use the "vbSpec" emulator for Windows available from:
"http://www.worldofspectrum.org" because it supports loading binaries directly into memory and it has a Timex TC2048 mode that allows programs that use the advanced video modes of the TS-2068 to run properly (recall Sprite Pack will generate programs for all the TS-2068's video modes). VbSpec's load binaries option asks for a filename and a memory address to load the binary (enter 32768). With the file in memory it can be run with "RANDOMIZE USR 32768".

To transfer the program to a real TS-2068 it must first be converted to tap format, which is an electronic representation of the contents of a tape. Z88DK has a tool that will do this for you. Run "\z88dk\support\zx\bin2tap myprog.bin 32768" to create the "myprog.tap" tape file. This file can be played out of the PC's soundcard and recorder on tape with the "Taper" utility (again available from World of Spectrum). Load this program from tape into your TS-2068 as you would any other program and then "RANDOMIZE USR 32768" to execute the program.

The one gotcha to all this is that the Z88DK libraries target the Spectrum and not the TS-2068. This almost doesn't matter, except for the few cases where the Z88DK libraries make calls into the Spectrum's ROM. For those few cases, the compiled program may need a Spectrum emulator to run properly on the TS-2068. Most programs generated will not require the emulator, fortunately. The Z88DK folks are just waiting for one of us to create a TS-2068 specific library to add to their collection.

This has been a brief introduction to the Z88DK compiler; there is still much to know but you have enough information to get going. It is important to know that Z88DK is being actively developed and as such has a few bugs (though very few!) and is being expanded to be more ANSI-C compliant. Between official releases, the latest bits of the compiler can be retrieved from the CVS repository at source forge and from:
http://www.algonet.se/%7Edennisgr/z88dk.htm
You should do the necessary update to fix a few bugs that have been corrected since version 1.5. If you don't know what CVS is, don't worry -- that's the subject of another article for another time.

Experienced C programmers: With the latest updates to Z88DK there is only three noticeable deficiencies that I am aware of. One is that function pointers cannot be prototyped. I solve this problem by declaring all function pointers as void* type. The second is that complicated types cannot by typedef'd. E.g.: it is okay to "typedef unsigned char uchar;" but it is not okay to "typedef struct {...} mystruct;" This also means pointers cannot be typedef'd as in "typedef void* pvoid;". The third is the lack of support for multidimensional arrays.

For more info, be sure to read the z88dk documentation in "C:\z88dk\doc", particularly the "zcc.htm" file.

SURPLUS ITEMS

TS Items (all prices include shipping cost). These are all complete with instructions, unless noted.

Items From RMG

- **TS-2068 States & Capitals (Timex cartridge) $7.00 (only 4 left)**
- **TS-2068 Spelling 1 (Timex cassette) $4.00**
- **TS-2040/Alphacom 32 printer paper $3.00/roll; 3 rolls/$7.50; 5 rolls/$10.00**
- **TS-1016 RAM pack $5.00**
- **TS-1000 User Manual $5.00**
- **The Ins and Outs of the Timex 1000 & ZX-81 $5.00 (book)**
- **TS-2068 BASICs and Beyond $5.00 (book)**
- **Computer Interfacing Techniques in Science $5.00 (book)**

The following TS-1000 individual software prices may seem high, but shipping for one is $1.92 First Class. A larger quantity would lower the effective individual price significantly by reducing the shipping cost, i.e., the price for one lot of any 6 below is $7.50. There is a significant quantity of these (except Gulp (3) & Super Math (2)).

- **TS-1000 Chess (Timex) $2.50**
- **TS-1000 Loan/Mortgage Amortizer (Timex) $2.50**
- **TS-1000 Super Math (Timex) $2.50**
- **TS-1000 States & Capitals (Timex) $2.50**
- **TS-1000 Red Alert (Softsync) $2.50**
- **TS-1000 Alien Invasion (Softsync) $2.50**
- **TS-1000 2K Games Pack (Softsync) $2.50**
- **TS-1000 Gulp (Mindware) $2.50**

So that I can track these through my organizational nightmare, please indicate in the Subject line of your email, "ZQA! Last Issue => RMG Items".

Magazines (each issue = $3)

SyncWare News: V2N1 through V3N5 (V2N4 has the address label cut out).
Time Designs Mag: V1N3, V1N4, V1N5, V1N6
Timex Sinclair User: #7 (last issue)
SYNC: Special Issue #2, V1N3, V2N4, V2N5, V3N2
SQ (Syntax Quarterly):V2N1
Ramblings V1N2 through V1N5

All are in pretty good condition, with the noted exception. There are scans of every one of these on the web site if you want to look at them.

From my private collection

- Non-working Sinclair QL (for parts?) $9.50 (computer only)
- Tandy 1000 EX, 640K RAM, internal (5-1/4") & external (3-1/2") disk drives, modem,
- CGA monitor, software manuals (very heavy, shipping would be expensive!) $25.00/or trade

The following are for trade only

- Australian Telecom Computerphone (QL clone)(no manuals, 220V power)

From Brazil

- CP-200 Computer (1000 clone by Prologica) 16K RAM, full keyboard, manual, 220V
- CP-200s Computer (1000 clone by Prologica) 16K RAM, full keyboard (this is new!) 220V
- TK-82c Computer (ZX80/MicroAce clone by Microdigital) 8K ROM, 2K RAM, 240v power supply, in original box with software

From China

- Power 3000 Computer (1000 clone) 2K RAM, 16K RAM pack, power supply

From Russia

- ZX Spectrum clone 48K RAM

Items I would trade for (or buy)

- **1959 Chevrolet El Camino**
- **Zebra/Timex FDD disk drive system: extra modules, manuals, software**
- **CP/M Software**
- **TS-1510 Cartridge Player & cartridges**
- **TS-2068 Pinball cartridge & box**
- **TS-2068 third party peripherals (disk interface, i/o boards, graphic boards, memory, etc., etc.)**
- **TS-1000 third party peripherals (see above)**
- **Sinclair ZX Interface 1 & microdrives**
- **TS-2068 twister boards**
- **TS-2068 cartridges (Timex or third party)**
- **TS Books, magazines, etc.**
- **TS Software**

Also, any TS item no longer needed would be willingly accepted for entrance into "The TIMEXsinclair Showcase", the history website of Timex computers. Hardware, software, books, documents, magazines, scraps of paper, ... anything is worth while. Nothing is EVER thrown out!

A lot of information has been added to the website recently (it's only taken 4 years to do a significant update). Right now ambition is high, so check in once in a while to see what's new! We may even get around to finishing the Spectrum Emulator cartridges started by Alvin Albrecht (and sat on by me) back in late 1999. They still need to have a capacitor soldered onto the boards and then tested. We've had 2 computer crashes in the past 18 months and have lost all the emails regarding this project (and

everything else archived, as well). Also a scan or picture of something you're doing (or something that's missing) would be a welcome addition to the site. Anyone having an idea for the site, or if anyone would like to help, let me know.

I will be retiring from my present position at the end of April and beginning a construction management consulting service, working out of my home.

JACK BOATWRIGHT
67325 FRYREAR RD
BEND OR 97701
Website: www.outlawnet.com/~jboatno4/index.html
Email: jboatno4@outlawnet.com (personal) or jbcs@outlawnet.com (business)
Telephone: 541-389-735

Present and Past Members

PAUL M ANDERSON
2226 DALEY DR
LONGMONT CO 80501

GERALD ANSON
2226 W SUNNYSIDE DR
PHOENIX AZ 85029-3564

ALVIN R ALBRECHT
1532 W 57TH AVE
VANCOUVER, BC
CANADA V6P 1T2

ROBERT L BARNETT
13726 RIVER FOREST DR SE
FT MYERS FL 33905-1820

DAVE BENNETT
1275 TIMBER VIEW DR
MECHANICSBURG PA 17055

M H BINSTOCK
1150 WINDERMERE DR
PITTSBURG PA 15218-1144

JACK BOATWRIGHT
27325 FRYREAR RD
BEND OR 97701

AL BOEHM
2501 ERMINE DR.
HUNTSVILLE, AL 35810

J A BOWERS
D 417 MCKNIGHT CIR
PITTSBURGH PA 15237-3538

WES BRZOZOWSKI
337 JANICE ST
ENDICOTT NY 13760

ALEX BURR
2025 O'DONNELL DR
LAS CRUCES NM 88001

JEFF BURRELL
4955 EMERSON AVE N
MINNEAPOLIS MN 55430

HOWARD CHEGWIDDEN
2 STEPHEN ST
DOVER NJ 07801

JAMES P CURRY
33 PULLMAN AVE APT4
ROCHESTER NY 14615

ROBERT CURNUTT
10400 TRUXTON RD
ADELPHI MD 20783-1118

ANDREW DANSBY
700-205 S W 16TH AVE
GAINEVILLE FL 32601

JOHN DONALDSON
835 FOXWOOD CIR
GENEVA IL 60134

BILL CABLE
PR 3 BOX 92
CORNISH NH 03745

GEORGE CHAMBERS
14 RICHOME CT
SCARBOROUGH ON M1K 2Y1
CANADA

LES COTTRELL
108 RIVER HTS DR
COCOA FL 32922-6630-14

JAIME A CRUZ-FIGUEROA
880 KRAMER RD
LILLINGTON NC 27546

CALLUM DAVIDSON
3119 WINBERRY DRIVE
FRANKLIN, TN 37064-6230

FRANK DAVIS
513 E MAIN ST
PERU IN 46970

WILLIAM DES LAURIERS
9926 KRAMER CT
INIANAPOLIS IN 46236-1647

DOMINO CUBES Z88
352 7TH AVE 15TH FLR
NEW YORK, NY 10001

DOUGLAS L DUNBAR
4515 5TH STREET NW
ALBUQUERQUE NM 87107

JACK DOHANY
627 BERA AVE
REDWOOD CITY CA 94061

DENNIS DONAHUE
25951 SUPERIOR
TAYLOR MI 48180

MARK DORINSON
PO BOX 351
FRANKFORT IL 60423

DAN ELLIOTT
COMPUTER CLASSICS
RR 1 BOX 117
CABOOL MO 65689

JOÃO ENCARNADO
RUA DO CASTELO Nº24 R/C
ESQUERDO PIRESCOXE
2695-247 STª IRIA DE AZOIA
PORTUGAL

RUTH FEGLEY
6000 IVYDENE TER APT 1E
BALTIMORE MD 21209-3547

MIKE FELERSKI
1284 BRUSHWOOD AVE
CINCINNATI OH 45224

CHARLIE FOX
7603 E FIRESTONE BLVD 129
DOWNEY CA 90241

LOUIS FLORIT
5445 SW 150 PLACE
MIAMI, FL 33185

JOHN FRANKE

23 PARKWOOD DR 201
YORKTOWN VA 23693-4819

ROBERT GILBERT
12A NATHAN RD
WALTHAM MA 02154

R ARTHUR GINDIN
821 STRALEY AVE 9
PRINCETON WV 24740

GLEN GOODWIN
7723 AVONWOOD CT
ORLANDO FL 32810

ROD GOWEN
14784 S QUAIL GROVE CIR
OREGON CITY OR 97045-8843

FERDINAND GUNTHER
1307 SHAW ST
MOSES LAKE WA 98837-3133

KENNETH R HARBIT
983 E KENOSHA AVE
FRESNO CA 93720-2175

ROBERT D HARTUNG
2416 N COUNTY LINE RD E
HUNTERTOWN IN 46748

CY HERRE
666 S BLUFF ST 802
S BELOIT IL 61080-2166

FREDERICK R HILL SR
136 LEXINGTON ST
WOBURN MA 01801

LEON HOWELL
6150 MONUMENT DR #2
GRANTS PASS OR 97526

PAUL HOLMGREN
5231 WILTON WOOD CT
INDIANAPOLIS IN 46254

GLENN HUFSTEDLER
4405 WINDSWEPT WY
LOVES PARK IL 61111-4429

ROD HUMPHREYS
10984 COLLINGS PL
DELTA B C V4C 7E6
CANADA

JOHN J IMPELLIZZERI
45294 UNIVERSAL CT
UTICA MI 48317-4941

WARREN JACKSON
11141 EDGEMERE TER
ROSCOE IL 61073

RICHARD A JELEN
11443 ISLAND RD
GRAFTON OH 44044

DAVID JOHNSON
3517 DUNEDIN DR 102
CHESAPEAKE VA 23321

TERRY N JONES
HC 65 BOX 2690
SPRINGFIELD WV 26763

JON J KACZOR
4568 WILLIAMSTON AVE
BROOKLYN OH 44144

HARRIET JOAN KEALY
PO BOX 1439
BRACKETTVILLE TX 78832

QUENTIN KENT
PO BOX 363
ALLENTOWN PA 18105

KEITH ELECTRONICS
224 NORTH GROVE ST
LOCK HAVEN, PA. 17745

EARL L KIELGASS
2015 E DUKE DR
TEMPE AZ 85283-2413

WAYNE KNAUST
2 PEAR TREE CT
ST PETERS MO 63376

JEFFREY A KUHLMANN
CMR/416 BOX D
APO AE 09140

WILLIAM KROSSNER
PO BOX 3047
DULUTH MN 55803-3047

BOB KWATER
1641 BLEDSOE DR
BELLBROOK OH 45305-1350

PHILLIP KWITKOWSKI
2106 DOVER LN
ST CHARLES IL 60174

DAVID E LASSOV
2590 N JORDAN DR
TUCSON AZ 85745-1132

DONALD LAMBERT
738 GUNNAR LN
FORSYTH IL 62535-8904

MEL E LA VERNE
103 ENDICOTT LN
OAK RIDGE TN 37830

DAVID LEECH
BYTE-BACK
536 LONG TER
LEESVILLE SC 29070

GARY LESSENBERRY
126 N CHAPEL ST
WAUKEGAN IL 60085-4120

PETER LIEBERT-ADELT
LÜETZOWSTRASSE 3
D-38102 BRAUNSCHWEIG
GERMANY

NESQLUG NEWS
ED KINGSLEY
16 HIGHLAND AVENUE
SAUGUS MA 01906

BILL MARRIOTT
11613 NE 97TH LN
KIRKLAND WA 98033

BILL MCKELVEY
612 MERCER AVE
SPRING LAKE HTS NJ 07762

JOHN MCMICHAEL
1710 PALMER DR
LARAMIE WY 82070

DEAN B MIKOLAJCZYK
4714 ARBOR DRIVE # 207
ROLLING MEADOWS IL 60008

HARRY E MILLER W1DRD
PO BOX 62
BERLIN MA 01503

SEYMOUR H MILLER
109-14 ASCAN AVE 1L
FOREST HILLS NY 11375

JOSE MORENO
1871 N GLADES DR APT 3
N MIAMI BEACH FL 33162

GREGORY M NEWKIRK
6321 KNIGHT AVE
LONG BEACH CA 90805-3840

JOHN OLIGER
THE J OLIGER CO
11601 WHIDBEY DR
CUMBERLAND IN 46229

KEVIN O'LEARY
BOX 172
MERIDEN NH 03770

DON OVIATT
5 MATTHEW CT
ARNOLD MO 63010-5126

GILLIAM Y PARRISH
11392 264TH ST
BEGGS OK 74421

NAZIR PASHTOON
1114 CATHEDRAL LN
MEMPHIS, TN 38103

JOHN B PEGRAM
1126 STRADELLA RD
LOS ANGLES CA 90077

LUKE PERRY
3409 NE 62ND AVE #187
VANCOUVER WA 98661

LEON PHELPS
PO BOX 116
KANEVILLE IL 60144

EDWIN N PHILLIPS
540 MARILEA CT
ORANGE CITY FL 32763

JOHN PORTERFIELD
5555 S BLACKSTONE #1
CHICAGO IL 60637

HARVEY RAIT
5 PERI LN
VALLEY STREAM NY 11581

JOE RAMPOLLA
8 MAGNOLIA LN
HANOVER PA 1733

WILF RIGTER
12388 88 AVE
SURREY BC V3W 7R7
CANADA

JOHN RISH
HOME ELECTRONICS SERVICE
5222 KAZEN DR
SAN ANTONIO TX 78219

ALBERT F RODRIGUEZ
AFR SOFTWARE
1605 PENNSYLVANIA AVE 204
MIAMI BEACH FL 33139

BILL RUSSELL

RUSSELL ELECTRONICS
RR1 BOX 539
CENTER HALL PA 16828

LARRY SAUTER
7747 W BRYN MAWR AVE
CHICAGO IL 60631

HUGH W SCRIVEN
235 E OXFORD ST
CHULA VISTA CA 91911

ROBERT G SCHIMKE
1005 WEST WIND CIR
PLACENTIA CA 92670-7043

LEE THORESEN
3065 E HICKORY LN
CRETE IL 60417

NEAL SCHULTZ
PO BOX 101
BUTLER WI 53007

JOHN J SHEPARD III
281 130TH ST
OGDEN IA 50212

ROBERT SHADE
3210 N BROAD ST
PHILADELPHIA PA 19140-5008

LOUIS A SIMON
34 CROWTHER AVE
BRIDGEPORT CT 06605-2504

THOMAS SIMON
615 SCHOOL AVE
CUYAHOGA FALLS OH 44221

FRANCINE SKLAR
HURLEYVILLE RD PO BOX 53
LOCH SHELDRAKE NY 12759

DAVID SOLLY
1545 ALTA VISTA DR 1402
OTTAWA ONTARIO K1G 3P4
CANADA

DANE L STEGMAN
26 MARSHALL AVE
AKRON NY 14001-1016

MARK STUEBER
7244 MECHANICSVILLE TPK
MECHANICSVILLE VA 23111

SUNSET ELECTRONICS
2254 TARAVAL ST
SAN FRANCISCO CA 94116

TIMOTHY SWENSON
2455 MEDALLION DR
UNION CITY CA 94587-1914

WALTER M SWENTKO MD
2311 30TH AVE S
MINNEAPOLIS MN 55406

ALEXANDER SWEITZER
RD 1 BOX 207
FAYETTE CITY PA 15438

ROBERT SWOGER
613 PARKSIDE CIR
STREAMWOOD IL 60107-1647

TEJ COMPUTER PRODUCTS
2405 GLENDALE BLVD 208
LOS ANGLES CA 90039

DON WALTERMAN
331 DRACE
ROCHESTER MI 48307

BARRY WASHINGTON
7044 CINDY LN
ANNANDALE VA 22003

DOUG WAGONER
E 4825 ST ANTHONY LN
POST FALLS ID 83854-8812

DON WALTERMAN
331 DRACE
ROCHESTER MI 48307

STUART W WALTON
31 PLEASANT ST
ROWLEY MA 01969-0913

WATCHARA CHANTANG
880 E FREMONT AVE #609
SUNNYVALE CA 94087

KEITH WATSON
41634 AMBERLY DR
MT CLEMENS MI 48038

BASIL WENTWORTH
1413 ELLISTON DR
BLOOMINGTON IN 47401

RUSSELL D WILES
826 S WEST AVE
SIOUX FALLS SD 57104-4641

BOB WILSON JR
PO BOX 25323
PORTLAND OR 97298

WESLEY J ZAPOTOCHNA
PO BOX 1
ROSELLE PARK NJ 07204

A Fast Well-Behaved Pattern Flood Fill

Alvin Albrecht

Recursion has always held a certain fascination for me because it seemed that it made very complex problems solvable with a simple, almost magical step. I have written about recursion before in a back issue of ZQA! so I won't be rehashing that material here, but a brief review is in order before we tackle the subject of flood fills.

A Review of Recursion

So what exactly is recursion? In simplest terms a recursive algorithm is one that calls itself. One class of recursive algorithms breaks a complicated problem into smaller, simpler pieces and then repeatedly applies itself to those pieces, breaking them down even further. Eventually, if everything goes well, you will wind up with many small problems that can be trivially solved. An example of a recursive algorithm of this class is a recursive solution to the Towers of Hanoi problem or the QuickSort sorting algorithm. Another class of recursive algorithms makes use of recursive backtracking -- a complex problem is solved by trying to solve it one step at a time. When a step is taken that cannot lead to a solution, the algorithm backs off and tries another step. Recursive algorithms in this class include solutions to the Eight Queens problem and the Knight's Tour problem. You can find recursive solutions to the Towers of Hanoi and Knight's Tour written in BASIC in a back issue of ZQA!.

For those of us without back issues of ZQA! handy, let's investigate the one recursive algorithm that everyone sees in Computer Science 101 -- the computation of a factorial. N! (N factorial) is defined as (N)x(N-1)x(N-2)x...x1 with N being a positive integer. 5! = 5x4x3x2x1 = 120, for example. 0! is defined as 1. A recursive solution might look like this:

```
int factorial(int n)
{
   if (n <= 1)
      return 1;
   return n*factorial(n-1);
}
```

Wait a minute, that's not a BASIC subroutine, that's a C function! Sinclair BASIC does not directly support recursion so a BASIC version of the above would need a bunch of artifacts decorating it, which would only obscure the point I want to make. Now that a decent C compiler is available for our T/S machines, I don't feel too guilty about sticking with the C. We make a call, asking to compute the factorial of N. If N <= 1 the answer is 1 (ignoring the case that N is negative which should throw an error). Otherwise the problem is too difficult to solve so it is broken into a smaller one: N times the factorial of (N-1).

That was easy, but it might be initially surprising to learn that this is a poor method for calculating factorials. To understand why, we'll need to pay closer attention to what is involved in a recursive call. The factorial function, as written above, needs to remember two things: the number N and where it was called from so that it can return there later. Compiling the above for a TS-2068 using the z88dk C compiler would reserve two bytes to store N and two bytes for the return address, for a total of four bytes. So the initial call to compute 5! would require four bytes to be reserved. But then factorial(5) would try to return "4*factorial(4)", causing another call to be made to factorial with N=4, requiring another four bytes. Factorial(4) would make a call to factorial(3), requiring a further four bytes, etc. In other words, to find the answer to factorial(5), the computer would make calls to factorial(5), factorial(4), factorial(3), factorial(2) and factorial(1). Factorial(1) returns 1 as the answer, factorial(2) returns 2*1 as the answer, factorial(3) returns 3*2 as the answer, factorial(4) returns 4*6 and factorial(5) returns 5*24. We say that factorial(5) has a recursion depth of 5 because a maximum of five instances of factorial will exist at any one time during its computation. With each instance needing four bytes to remember its value of N and return address, factorial(5) requires 5*4=20 bytes of memory to compute the result. More generally we can say that the recursion depth of factorial(N) is N and 4*N bytes are required to compute the result.

Even on our small 64K machines, that doesn't seem to be a lot of memory and really it isn't. Things get worse if you try to compute something like 69! which would seem to require 276 bytes in the recursive solution. I say "seem to" because we would need a new large variable to hold the result in each recursion step -- 69! requires 41 bytes to hold its value precisely! Switching to a floating point representation would reduce that to four bytes (at the expense of precision), compared to the two bytes we've assumed (the result in each step is held in the HL register pair, a consequence of how the z88dk C compiler does things). With the way our subroutine is written now we couldn't compute anything more than 8! and therefore memory usage is never really an issue. Other recursive algorithms may have a recursion depth in the thousands with dozens of bytes needed for each instance. Then the matter of memory is significant, and, indeed, you will see one such example shortly.

Another problem with the factorial recursive solution in our case is runtime. It takes time to set up calls and return from them -- these actions translate directly into pushes and pops on the z80's stack. Compare this recursive solution to the "regular" iterative solution below:

```
int factorial(int n)
{
   int fact, i;

   fact = 1;
   for (i=2; i<=n; i++)
      fact = fact*i;
   return fact;
}
```

We need two bytes for "fact", two bytes for "i", two bytes for "N" and two bytes for the return address = 8 bytes total no matter what "N" is. There are no calls to set up, just a for loop. This version will be faster and use up a small, fixed and predictable amount of memory. It is superior to the recursive solution in every way.

So what is the conclusion of all this discussion? You must use recursion with care. It can be a panacea to solve many very difficult problems, but you must be fully aware of how much memory will be required and the runtime necessary in comparison to an equivalent iterative solution. The Towers of Hanoi solution in a back issue of ZQA! has a maximum recursion depth of 64 (for 64 disks) and the Knight's Tour has a recursion depth of 64 (the number of squares on a chess board), very manageable numbers. You can see in both of those BASIC solutions that an array of those sizes was allocated to remember the state in each recursive step.

A Recursive Flood Fill Algorithm

So what has all this got to do with flood filling? It turns out that the obvious approach to filling an arbitrary region involves a recursive solution. And the recursive solution is a bad one.

In comp.sys.sinclair, Geoff Wearmouth shared a BASIC subroutine from an early '80s type-in magazine that would fill an arbitrary area on screen bounded by a solid pixel boundary. Here it is:

```
5 REM AUTHOR UNKNOWN
10 CIRCLE 128,88,4
20 LET x=100 : LET y = 100 : REM START
POINT
30 GO SUB 1000 : STOP
1000 PLOT x,y
1010 IF NOT POINT(x+1,y) THEN LET x=x+1 :
GO SUB 1000 : LET x=x-1
1020 IF NOT POINT(x-1,y) THEN LET x=x-1 :
GO SUB 1000 : LET x=x+1
1030 IF NOT POINT(x,y+1) THEN LET y=y+1 :
GO SUB 1000 : LET y=y-1
1040 IF NOT POINT(x,y-1) THEN LET y=y-1 :
GO SUB 1000 : LET y=y+1
1050 RETURN
```

The main subroutine begins at line 1000 and the algorithm used is a recursive one (did you spot the "GOSUB 1000" statements in the subroutine itself?) Filling an arbitrary region is not a trivial problem so a recursive approach is taken to whittle away all the complexity. The fill subroutine above plots the current point (the PLOT on line 1000) and then tries to move in all four directions away from the point. Before each move, it checks to see if the point is already black, indicating a boundary. If not, it considers it a valid move and tries to fill from that point by calling itself with the new pixel coordinate in (x,y).

Earlier I mentioned that Sinclair BASIC does not directly support recursion. The reason it doesn't can be seen in this fill program. Each run through the subroutine at 1000 expects to have its own private copy of (x,y). The value of (x,y) at 1010 must be the same value at line 1020, 1030 and 1040 in order for the program to work. But there are one or more "GOSUB 1000" calls in the middle, which themselves require new values of (x,y) and which will themselves change (x,y). Sinclair BASIC has only one copy of these variables which must be shared by each recursive call. A recursive C call would give each "GOSUB" a private copy of (x,y) on the stack independent of all other "GOSUBs". Not so in Sinclair BASIC. So the problem is, after each "GOSUB 1000" in the fill subroutine, how do we make sure that our own (x,y) has not changed? In the above code, the solution is simple. We promise that when "GOSUB 1000" returns, the value of (x,y) is not changed from what it was when "GOSUB 1000" was executed. In line 1010, for example, x is increased by one before a call to "GOSUB 1000". Because of our promise (the jargon calls such a promise an "invariant") we know that when the GOSUB returns, x will be one larger than what it was at the beginning of line 1010. So to get x back to where it was, decrease by one and everything will be fine for the next line. Before the routine returns in line 1050 we know that (x,y) has not changed from its initial state in line 1000. That's the subroutine keeping its promise.

The kind of fill algorithm this program uses is formally called a flood fill because the fill "floods away" from the initial point in all directions. Other fill algorithms exist, but this one is both easy to understand and capable of filling any arbitrary region without restrictions. Earlier I hinted that a recursive solution to the flood fill problem is a bad one. I'll leave that thought here and come back to it later when we've looked at a couple of machine code implementations of the algorithm. For now, realize that each "GOSUB" requires the TS-2068 to remember a return line number (two bytes) and then consider what the recursion depth might be for a 256x192 resolution blank screen (hint - you wouldn't be far off if you just multiplied 256 and 192 together!).

If you typed in the BASIC program and ran it, you'd realize that it is mighty slow. Any useful fill utility will need to be written in machine code. To do that, we will need to review the structure of the TS-2068's display file.

Display File Organization

The TS-2068's display file is where all the screen information is stored. The SCLD chip constructs the TV display by reading the information stored there. The display file is termed "memory-mapped" because the storage exists in the z80's memory space, from address 16384 to 22527 (not counting the colour attributes for each character). If you poke values into those addresses you will see the display change. In the TS-2068's other display modes (dual screen, hi-colour, hi-res) more areas of memory are used to hold the display. In this article, we'll only concern ourselves with the default 256x192 mode.

A pixel display occupying 16384 to 22527 reserves 6144 bytes to store all the screen information. The TS-2068 has a resolution of 256*192 = 49152 pixels. How do we cram information about 49152 pixels into 6144 bytes? Well, each pixel can be represented by one bit - either one or zero, on or off. Cramming 8 pixels into a byte, we'd need 256*192/8 = 6144 bytes. Problem solved!

A simple way to organize the display might have pixels 0..7 for the top line of the display stored at address 16384, pixels 8..15 at address 16385, ... pixels 248-255 stored at address 16415. The next pixel line would follow with pixels 0..7 of line 1 at address 16416, and so forth for all 192 lines on the screen. This is indeed how the TV draws its display, left to right, top to bottom. But the display organization was chosen to optimize the printing of characters so it's not done in this simple manner. To see evidence of this, try this short program:

```
10 FOR z=16384 TO 22527
20 POKE z,255
30 NEXT z
```

If you run this program you will witness how the display file is organized. On the largest scale you will notice that the display is divided into three parts (called blocks). First the top block is filled, then the second and finally the third. Each block is further divided into eight character lines. Each of these lines is divided into eight scan lines. The first scan line for all character lines in a block is filled, followed by the second scan line for all character lines, all the way to the final eighth scan line. Each scan line itself is composed of 32 horizontal bytes with each byte holding eight pixels.

This organization sounds complicated but it really isn't that bad if some thought is applied to it. By paying attention to how the display is built up in increasing byte order (reread the paragraph above), we can construct a screen address given block, character line, scan line and column as follows:

FIGURE 1. Screen Address Organization

0	1	0	B	B	S	S	S	L	L	L	C	C	C	C	C

Where:

```
BB = screen block, 0..2
SSS = scan line, 0..7
LLL = character line, 0..7
CCCCC = horizontal byte / character, 0..31
```

While observing the BASIC program, you'll notice that the horizontal column changes the fastest. There are 32 columns, requiring 5 bits to represent those. They increase the fastest so they appear in the bottom 5 bits of the 16-bit address. The next fastest thing that changes is the character line. There are 8 lines in each block, requiring 3 bits to represent them. These 3 bits appear next to the column bits. Next, in order of fastest changing, are the scan lines (8 of them requiring 3 bits) followed by the block (3 of them requiring 2 bits). The display starts at address 16384 (0x4000) so we add that to our 16-bit address. This is responsible for the lone '1' you see in figure 1.

The character position row = 10, column = 12 is located in block 1 (the second block since it holds the second third of the display, rows 8..15), line 2 (the third character line in this block -- rows 8, 9, **10**), scan lines 0 (top) through 7 (bottom) for the full character square, and column 12. This leads to a screen address that looks like:

0	1	0	0	1	S	S	S	0	1	0	0	1	1	0	0

With various values of SSS:

SSS	**0**	**1**	**2**	**3**
Screen	484C	494C	4A4C	4B4C
Address	18508	18764	19020	19276

To print a letter 'A' at (10,14), poke the appropriate values into memory at these addresses:

```
POKE 18508,BIN 00000000
POKE 18764,BIN 00111100
POKE 19020,BIN 01000010
POKE 19276,BIN 01000010
POKE 19532,BIN 01111110
POKE 19788,BIN 01000010
POKE 20044,BIN 01000010
POKE 20300,BIN 00000000
```

At this point, you may realize why UDGs and printed characters are 8x8 pixels in size. There are 8 vertical scan lines in each character line and there are 8 pixels packed into a byte. But you may not realize why this particular display file organization speeds up character printing. If you back up and look at the screen addresses computed above, you'll notice that each scan line is separated by 256 bytes (you could also see this from figure 1). In assembly language, an address is held in a register pair, like HL. Adding 256 to an address is a quick matter of incrementing the most significant register, in this case H with the "INC H" instruction. That's all it takes! Moving horizontally to the right one character position involves adding one to the screen address, which can be done just as quickly with "INC L". You can't get any faster than that. In fact, this display file organization was patented by Sinclair's Richard Altwasser back in 1982 (visit http://... to see the patent).

That's all fine and good but we still haven't managed to easily map a pixel coordinate to a screen address. Here's how we do it:

FIGURE 2. Mapping Pixel Coordinates to Screen Addresses

	X								Y						
C	C	C	C	C	T	T	T	B	B	L	L	L	S	S	S

The thought process that led to figure 2 is similar to the previous one. The X coordinate is more or less obvious: there are 32 columns horizontally (5 bits) with each column containing 8 pixels (requiring 3 bits). The pixel position within a byte (0..7) changes fastest as we move horizontally so it appears as "TTT" in the least significant bits of X. For the Y coordinate, the fastest changing item as we move from the top of the screen to the bottom is the scan line, followed by the character line, followed by the block.

Given an X coordinate in the range 0-255 and a Y coordinate in the range 0-192, convert them to binary as in figure 2 and reassemble the bits as in figure 1. For example, pixel coordinate (x,y) = (133,67) in binary is (1000 0101, 0100 0011) with CCCCC=10000, BB=01, LLL=000, SSS=011 according to figure 2. Moving the bits around to the form in figure 1 gives an address of "0100 1011 0001 0000" or 19216 in decimal. The bits "TTT" in the X coordinate do not appear in figure 1. They identify which bit within the screen byte corresponds to the individual pixel. "0" corresponds to the leftmost bit and "7" corresponds to the rightmost; in this case it's 5. To plot the pixel (133,67) we could simply "POKE 19216,BIN 00000100"

4	5	6	7
4C4C	4D4C	4E4C	4F4C
19532	19788	20044	20300

If this had to be done by hand for each pixel, it would get tedious quickly. Here's a short machine code routine that follows this procedure:

```
; Get Screen Address
;
; Returns the screen address and pixel mask corresponding
; to a given pixel coordinate.
;
; enter: a = h = y coord
;        l = x coord
; exit : de = screen address, b = pixel mask
; uses : af, b, de, hl
.SPGetScrnAddr
   and $07      ; A = 00000SSS
   or $40       ; A = 01000SSS
   ld d,a       ; D = 01000SSS
   ld a,h       ; A = Y coord = BBLLLSSS
   rra
   rra
   rra          ; A = ???BBLLL
   and $18      ; A = 000BB000
   or d         ; A = 010BBSSS
   ld d,a       ; D = 010BBSSS top 8 bits of address done
   ld a,l       ; A = X coord = CCCCCTTT
   and $07      ; A = 00000TTT
   ld b,a         ; B = 00000TTT = which pixel?
   ld a,$80     ; A = 10000000
   jr z, norotate     ; if B=0, A is the right pixel so skip
.rotloop
   rra          ; rotate the pixel right one place B times
   djnz rotloop
.norotate
   ld b,a       ; B = pixel mask
   srl l
   srl l
   srl l        ; L = 000CCCCC
   ld a,h       ; A = Y coord = BBLLLSSS
   rla
   rla          ; A = LLLSSS??
   and $e0      ; A = LLL00000
   or l         ; A = LLLCCCCC
   ld e,a       ; E = LLLCCCCC
   ret          ; DE = 010BBSS LLLCCCCC, the screen address!
```

The subroutine is called with A=H=Y coordinate and L=X coordinate and we get the screen address in DE and the pixel mask in B on the way out. If we ORed B into (DE), we could plot the pixel. If we ANDed the complement of B into (DE), we could unplot the pixel and if we ANDed B with (DE) we could test whether the pixel was set.

This subroutine is great for calculating a screen address corresponding to a pixel position from scratch, but you'll notice that it is rather lengthy and therefore slow, in a relative sense. Frequently you'll be plotting a pixel and then plotting many more nearby, possibly a single pixel away. For example, in the process of drawing a line, the initial point is plotted and then succeeding points above, below, to the left or right are plotted. We could handle the drawing of the line as plotting many individual pixel points, calling the above subroutine to compute the screen address for every pixel, but that would be much slower than working directly on the screen address to move up, down, left and right from a current pixel position.

Let's investigate further to substantiate that claim. Given a screen address in HL and a pixel mask in B, how would one move left one pixel? Here's the necessary code:

```
; hl = screen address, B = pixel mask
.left
   rlc b
   ret nc
   dec l
   ret
```

The pixel mask is rotated left one bit. This will be a valid pixel position unless B was already at the leftmost pixel position in the screen byte (i.e. B=1000 0000). The "RLC B" instruction will set the carry flag in that case and leave B=0000 0001. We use the no carry flag to return early if the new screen address and mask is valid, otherwise we update the column position one character to the left (the "CCCCC" portion of the screen address in figure 1). The value of B at this point is 0000 0001, correctly masking the rightmost pixel in the new screen byte to the left of the old one. These four instructions are clearly quicker than rerunning the screen address subroutine. The right pixel movement is similar, substituting "rrc b" for "rlc b" and "inc l" for "dec l". Notice that these subroutines don't check if we run off the edge of the screen on the left or right side.

To move up a pixel, we need to move up a scan line (decrease "SSS" in figure 2). If we wrap past the first scan line (0), we need to decrease the character line. If we move past the first character line (0), we need to decrease the block. Here is the necessary code:

```
; hl = screen address
.SPPixelUp
   ld a,h          ; A=H=010BBSSS
   dec h           ; decrease SSS
   and $07              ; if SSS was not originally 000
   ret nz          ; we're done
   ld a,$08             ; otherwise SSS=111 (correct)
   add a,h         ; and we fix BB in H (one was subtracted)
   ld h,a
   ld a,l          ; A=X coord=LLLCCCCC
   sub $20         ; decrease LLL
   ld l,a
   ret nc            ; if no carry, LLL was not originally 000, okay
   ld a,h          ; otherwise LLL=111 now, that's okay
   sub $08          ; but need to decrease screen block
   ld h,a
   ret
```

This subroutine derives a lot of speed by minimizing the number of instructions executed in the most common cases. For example, 7 out of 8 times, only the first four instructions will be executed. 7 out 64 times, the first 11 instructions will execute and the rest of the time (1 out of

64) all the instructions will execute. This makes the subroutine much quicker than one would initially guess by looking at the size of the code. The PixelDown subroutine is similar but is not shown here. All these pixel movement routines are reprinted in full in the floodfill listings elsewhere in this article.

That's enough information to have a first crack at a machine code version of the BASIC flood fill routine.

Machine Code Flood Fills

Figure 3 contains a direct conversion of the BASIC flood fill we saw earlier. No optimization has been done but it has been improved slightly to check for moving across the screen boundary. Type in the associated BASIC listing to see it in action.

We have managed to speed things up considerably by moving to machine code, but there are still a couple of improvements that can be made. First, we compute screen addresses for every single point plotted. Since we always move up, down, left or right from the current pixel we could speed things up by avoiding this computation as discussed earlier. The other optimization we can make is to plot a full 8 pixels at a time rather than one. What? Recall that each screen byte holds eight pixels. Why fiddle with it eight times to plot eight pixels in it when we could plot all 8 pixels at once with a single write of a whole byte?

The secret to plotting multiple pixels at once is the bytefill subroutine. It operates directly on a screen address and pixel mask, exactly what we will have available now that we have decided not to compute the screen address for every single pixel plotted.

```
; hl = screen address
;  b = incoming pixel mask
.Bytefill
   ld a,b          ; get pixel mask
   xor (hl)            ; zero out incoming
pixels that
   and b            ; run into set pixels in
display
   ret z           ; if no pixels left, ret
.bfloop
   ld b,a          ; b = incoming pixels
   rra             ; expand incoming pixels
   ld c,a          ; to the right and left
   ld a,b          ; within byte
   add a,a
   or c
   or b               ; a = incoming pixels
wiggled
   ld c,a          ; save in c
   xor (hl)          ; zero out pixels that
run into
   and c           ; set pixels on display
   cp b               ; have pixels changed
from last loop?
   jp nz, bfloop     ; keep going until
incoming does not change
   or (hl)
   ld (hl),a       ; fill byte on screen
   scf              ; indicate that this was
a viable step
   ret
```

Bytefill is called with a screen address in HL and a pixel mask containing all the "incoming" pixels. The incoming pixels are those pixels from where the flood fill grows in the current screen byte. Previously the flood fill always grew from a single point but not anymore. The origin of the incoming byte will be clear while perusing the second flood fill listing in figure 4.

The Bytefill routine takes the inconing pixels and "wiggles" them to the left and right, trying to grow them into blank spaces within the screen byte. It does this until no more growth is possible within the screen byte. It then plots all those pixels and returns.

Putting these ideas into action produces figure 4, a byte-at-a-time flood fill routine. Type in the BASIC listing to see it in action. This routine is blazing fast; you will not see anything quicker. But, and this is a big but, there is a major flaw in the recursive algorithm that it shares with all the previous fills we have seen so far: the recursion depth is huge.

Consider a flood fill from the bottom left corner of a blank screen. According to figure 4, the first thing that is done is the fill of the entire screen byte in the bottom left corner. Then a move to the right is made and its byte is filled in a recursive call to "fill". Followed by another right move and fill, then another, until we hit the right edge of the screen. A right move is not possible from the right edge of the screen so a left move is tried from there. That is unsuccessful because it was just filled. An up movement from the right edge is tried, successfully. Now we are at the right edge, one pixel up from the bottom of the screen. The filler fills in the byte and tries a right movement. That's not possible so it successfully tries a left movement. If this is carried on you'll notice that, from the bottom left corner of a blank screen, the screen is filled alternately from the left to the right and then from the right to the left as the fill line moves one pixel higher for each scan line filled. You may have noticed that not a single return instruction is executed during the entire screen fill. That is a problem. Each call to fill puts at least 2 bytes on the stack to remember the return address. Since no returns are made for all screen bytes on the screen, there are 6144 calls made to fill without a single return. That's a recursion depth of 6144! Since at least 2 bytes are saved on the stack for each recursive call, at least 12288 bytes are needed to complete the fill. The situation can be much worse, however. In the worst case movement (up or down) 6 bytes are saved on a recursive call (two for bc, two for hl and two for the return address). We may need up to 36864 bytes to fill the screen! The truth is somewhere in the middle. Notice that by moving from a pixel fill to a byte fill we have reduced the depth of recursion by a factor of eight. Still this amount of memory usage is unacceptable for most applications. How can you use a flood fill in your own programs if it's going to need most of the available memory to complete?

This recursive fill is an example of a depth-first algorithm. It fills an area by going as deep as possible into the area (one call begets another call begets another, etc. without returning). The result is a recursion depth that

is proportional to the area of the screen to be filled. We can rescue the situation by considering another algorithmic approach, known as a breadth-first algorithm. Instead of going as deep as possible into an area, try going wide first. This sounds like a lot of metaphysical talk of questionable value, but the terms "depth-first" and "breadth-first" are bonafide jargon that is used to describe the solution behaviour of many kinds of algorithms.

A breadth-first approach to a flood fill would, instead of going as far into a screen as possible, try to fill all points in the immediate area first. From a starting point, all pixels to the immediate left, right, top and bottom are filled. Then for all those adjacent pixels, their immediate neighbours are filled, etc. The savings comes from a key observation: once the immediate neighbours of a pixel are filled, there is no need to remember (come back to) the current pixel. Its information can be forgotten. This was not possible in the previous depth-first approaches. As will be seen later, the breadth-first approach will have a "recursion depth" proportional to the circumference of the area being filled, a significant savings.

Figure 3. Pixel Coordinate Flood Fill

```
; Flood Fill Version 1
; H = Y coord 0..191, L = X coord 0..255
.flood1
   push hl                          ; save (x,y)
coordinate
   ld a,h                           ; GetScrnAddr
requires A=H
   call SPGetScrnAddr      ; compute screen
address
   pop hl                    ; restore (x,y) in
HL
   ld a,(de)               ; byte on screen
   and b                        ; check if this
pixel is set
   ret nz                          ; if so, hit
boundary so ret
   ld a,(de)               ; get screen byte
   or b                    ; set this pixel
   ld (de),a               ; plot it on screen
.right
   inc l                     ; move pixel coord
right
   call nz, flood1         ; if no wrap 255->0
   dec l                   ; restore x coord
.left
   dec l                     ; move pixel coord
left
   ld a,l
   inc a
   call nz, flood1         ; if no wrap 0->255
   inc l                   ; restore x coord
.up
   dec h                     ; move pixel coord
up
   ld a,h
   inc a
   call nz, flood1         ; if no wrap 0->255
   inc h                   ; restore y coord
.down
   inc h                     ; move pixel coord
down
   ld a,h
   cp 192
   call c, flood1          ; if less than 192
   dec h                   ; restore y coord
   ret
```

Figure 4. Byte-At-A-Time Flood Fill

```
; h = y coord, l = x coord
.flood2
   ld a,h
   call SPGetScrnAddr     ; b = pixel mask
   ex de,hl                      ; hl = screen
address
.fill
   call Bytefill             ; wiggle around
incoming pixel mask
   ret nc                        ; if incoming
pixels hit boundary, ret
.up
   push hl                       ; save screen
address
   call SPPixelUp            ; move up one
pixel
   jr c, offscreen1
   push bc                  ; save pixel mask
   call fill                ; try to fill from
new screen position
   pop bc                   ; moving up, pixel
mask remains same
.offscreen1
   pop hl
.down
   push hl
   call SPPixelDown
   jr c, offscreen2
   push bc
   call fill
   pop bc
.offscreen2
   pop hl
.right
   bit 0,b                  ; if first pixel
in mask set, try right
   jr z, left
   inc l                    ; move right one
byte
   ld a,l
   and $1f
   jr z, offscreen3
   push bc                       ; save current
pixel mask
   ld b,$80                      ; new incoming
mask = leftmost pixel set
   call fill                ; fill from new
screen position
   pop bc
.offscreen3
   dec l
.left
   bit 7,b
   ret z
   ld a,l
   and $1f
   ret z
   dec l
   ld b,$01
   call fill
         ret
```

Unclassified Ads

QL, Spectrum, ZX-81 and Z88
www.members.tripod.com/hes_computing/hes1.html
E-Mail 74601.1535@compuserve.com
Phone 210 661-4376

Home Electronics Service
John R. Rish
5222 Kazen Dr.
San Antonio TX 78219 USA

The John Oliger Co.
11601 Widbey Dr.
Cumberland IN 46229
The John Oliger Floppy Disk System
FOR THE TS-2068
joliger@mindspring.com

NASQLUG

North American Sinclair QL User Group

Al Boehm
2501 Ermine Drive
Huntsville, AL 35810
albertboehm@juno.com

Kevin O'Leary
Box 172
Meriden, NH 03770.
olearyk@cyberportal.net

WOOD & WIND
ARCHIVE Based QL Software
Bill Cable
Wood & Wind Computing
RR3 BOX 92
Cornish NH 03745 USA
Phone (603) 675-2218

Domino Cubes

Z88

Hardware & Software
352 7th Ave. 15th Flr.
New York, NY 10001
Phone 212 631-7563
Domino.cubes@excelsior.net

QL Hacker's Journal

Supporting All QL Programmers

Timothy Swenson, *Editor*
2455 Medallion Dr.
Union City, CA 94587-1914
swensontc@geocities.com
http://www.geocities.com/SilconValley/Pines/5865/

ZX-TEAM MAGAZIN
Peter Liebert-Adelt
LUETZOW STR 3
D-38102 BRAUNSCHWEIG
GERMANY
Email: p.liebert@t-online.de
http://home.t-online.de/home/p.liebert/zx-team.htm

Software for the Timex\Sinclair Computer
MC, VISA, American Express. Phone 717-748-1747

Keith Electronics
224 North Grove St
Lock Haven, PA. 17745

Surplus T/S Inventory

JOHN J SHEPARD III
281 130th ST
OGDEN IA 50212
< jshepard@wccta.net >

Mostly QL & TS-2068

JACK BOATWRIGHT
67325 FRYREAR RD
BEND OR 97701
< jboatno4@outlawnet.com >

Mostly ZX-81/TS-1000 & TS-2068

Sinclair E-Mail List

Anderson, Paul	pandersn@peakpeak.com
Anson, Gerald	jerrya@aztec.asu.edu
Barker Robin	robin@di-ren.demon.co.uk
Bennett, Dave	dave975@att.net
Bill McKelvey	mckelveyw@delphi.com
Boehm, Al	albertboehm@juno.com
Boehm, Bill	boehm@plh.af.mil
Burrell, Jeff	jburrell@endocardial.com
Cable, Bill	cable@cyberportal.net
Carpio, Juan	juanchuscar@yahoo.com
Castro Antonio	castrox@portoweb.com.br
Catotti, Christopher	kd4ace@compuserve.com
Chambers, George	gfchamb@pathcom.com
Chantang, Watchara	w.chantang@attbi.com
Collins, Bill	bcollins@home.ifx.net
Cottrell, Les	jacottrell@cfl.rr.com
Cruz-Figueroa, Jaime	cruzfiguer@aol.com
Dansby, Andrew	adansby@atlantic.net
Davidson, Callum	callumdavidson@hotmail.com
Davis, Frank	fdavis@iquest.net
Delhez, Carlo	carlo@spase.nl
Donaldson, John	goodolejohn@avenew.com
Dorinson, Mark	74200.257@compuserve.com
Dunbar, Douglas	dldunbar@prodigy.net
DuPuy, James	dupuy@pipeline.com
Encarnado, João	encarnado@netcabo.pt
England, William	wengland@iname.com
Fegley, Ruth	ruth.fegley@worldnet.att.net
Feng, Al	alfeng@juno.com
Fink, Mike	domino.cubes@excelsior.net
Fink, Mike	domino.cubes@pointblank.com
Firshman, Tony	tony@firshman.demon.co.uk
Florit, Louis	florit@unixville.com
Franke, John	j.m.franke@larc.nasa.gov
Ganger, Gary	gangerg@dma.org
Gilbert, Robert	weena@netzero.net
Gillespie, Doug	aa431@cleveland.freenet.edu
Girnius, William	girnius_w@bls.gov
Goodwin, Glen	glenatacme@aol.com
Gowen, Rod	aw723@osfn.org
Haberly, Duncan	duncan@military.com
Harbit, Ken	krh03@cvip.fresno.com
Harris, Paul	plh@frsl5.f9.co.uk
Hartung, Bob	revrdhtp@netscape.net
Henderlight, Mike	mikehend@microsoft.com
Herre, Cy	cyherre@aol.com
Holmgren, Paul	paulholm@indy.net
Horton, Will	willhort@aol.com
Humphreys, Rod	rodh@pacificcoast.net
Impellizerri, John	jimpellizerri@compuserve.com
Jaap, Matthias	matthias_jaap@hhs.hh.schule.de
Jonas, Mike	mjonas@bbn.com
Jones, Dilwyn	dilwyn.jones@dj.softnet.co.uk
Jones, Terry	tjones@iname.com
Kaczor, Jon	jazkaczor@aol.com
Kahale, Abed	akahale@juno.com
Kealy, Harriet Joan	hjkealy@rionet.org
Kenny, Larry	larken@storm.ca
Kingsley, Ed	edk4@aol.com
Knyszek, Theodore	thirteenth@worldnet.att.net
Kondrac, Mike	mkandrac@aol.com
König, Urs	urs.koenig@agrodata.ch
Kwitkowski, Phillip	pkwitkowski@hotmail.com
Lambert, Donald	dslambert@emailmsn.com
Lancaster, Garry	dharkhig@delphi.com
Lanciault, Francois	francois.lanciault@energies.alstom.ca
LaVerne, Melvin	mlaverne@usit.net
Lebowitz, Dave	dkl@dpliv.com
Lessenberry, Gary	gl743@aol.com
Liebert-Adelt, Peter	peter@zx81.de
Liebert-Adelt, Peter	p.liebert@t-online.de
Malloy, Bob	74776.2342@compuserve.com
Matthias, Jaap	mjaap@atari-computer.de
McBrine, William	wmcbrine@covad.net
McKelvey, William	mckelveyw@delphi.com
Merz, Jochen	jmerz@t-online.de
Mikolajczyk, Dean	deanm97493@aol.com
Miller, Seymour	seymil@delphi.com
Muth, Bob	bobkeeper1@aol.com
Nachbaur, Fred	fnachbaur@netscape.net
Norton, Gary	gnorton@rsacc.net
Norton, Gary	gnorton@world.std.com
O'Leary, Kevin	olearyk@cyberportal.net
Parrish, Gil	gil.parrish@abanet.org
Payne, Josh	joshpayne@bigfoot.com
Pazmino, John	john.pazmino@moondog.com
Perry, Luke	doidy1@juno.com
Perry, Russ Jr	slapdash@enteract.com
Rampolla, Joe	jprampolla@blazenet.net
Rigter, Wilf	wilf.rigter@powertechlabs.com
Rish, John	74601.1535@compuserve.com
Sauter, Larry	sauter41738@msn.com
Shepard, Jay	jshepard@wccta.net
Simon, Thomas	73177.333@compuserve.com
Skapinski, Thomas	tskapins@juno.com
Solly, David	k_david_solly@hotmail.com
Stegman, Dan	danesteg@juno.com
Swenson, Tim	swensont@lanset.com
Swentko, Wally	wswentko@maroon.tc.umn.edu
Swoger, Robert	rswoger@aol.com
Taylor, Jeff	jetaylor@mdrobotics.ca
TEJ Computer	tej@jps.net
Thoresen, Jeff	74200.257@compuserve.com
Waldman, Stephen	brogine@hotmail.com
Walterman, Don	walterm@ix.netcom.com
Watson, Keith	keith_watson@juno.com
Webster, Robert	rwebs1@netzero.net
Zimmerman, George	gzimmer928@aol.com

ZQA! 1991 to 2003

VOLUME 1 NUMBER 1 Spring 1991	
SNUG is Dead! - Long Live T/SNUG	Bob Swoger
From The Chairman's Disk - SNUG!	Don Lambert
Oliger Proposes Détente With LarKen	Don Lambert
Stepping Rate Fix (Disk drives)	Swoger & L. Kenny
VOLUME 1 NUMBER 2 Summer 1991	
Input/Output - Censorship, Mailing, N/L Exchange, Mile High, Bill Ferrebee	Bob Swoger
Reviews - QL Survivor's Source Book	Dyl & Frank Davis
Reviews - International QL Report	Bob Dyl
T/SNUG, QUANTA, and PD-QL Software	Al Feng
Converted TS-1016 RAM That Works	Anthony Farrell
Cassette LOAD/SAVE Problems Solved I	Don Lambert
Solar System Review – Computers/Tech.	Joan Kealy
VOLUME 1 NUMBER 3 Fall 1991	
Input/Output - Gill Parrish, Louis Eisen, Joan Kealy, Bill Ferrebee and Rod Humphreys	Bob Swoger
Reviews - 1 Meg Printer Buffer from Technologic Systems	Bob Swoger
How To Replace TS-1000 Parts Cheap	Gill Parrish
Cassette LOAD/SAVE Problems Solved II	Don Lambert
QLUSTer_LITE	Al Feng
VOLUME 1 NUMBER 4 Winter 1992	
Input/Output - G. Chambers, W. Harmer, R. Wisti, H. Scriven, B. Allen and R. Madaris	Bob Swoger
Cassette LOAD/SAVE Problems Solved III	Don Lambert
Clean Screen Please TS-2068	Bob Swoger
A Short History of The Z-88	Dave Bennett
Transfer MS-DOS Files to the TS-2068	George Chambers
QL Directory to Printer PDS DOC	Butch Wienberg
ZX-91 10 Years Later No.1 & No.2	André Baune
VOLUME 2 NUMBER 1 Spring 1992	
Input/Output - W. Jackson, M. Kendoll, R. Shade, I. Zachev and J. Shepard	Bob Swoger
Cassette LOAD/SAVE Problems Solved IV	Don Lambert
An Updated ZX-81	Don Lambert
Bench Marking The ZX-81	Edward Snow
Driving Your Tax Threshold	Joan Kealy
TAX-I-QL TIP	Al Feng
ZX-91 10 Years Later No.3	André Baune
VOLUME 2 NUMBER 2 Summer 1992	
Quanta-Gate?	Al Feng
Input/Output - K. Krack, G. Hufstedler, F. Sklar, C. Fox , A. Baune D. Lambert, J. Rampolla, C. Kereluck, P. Anderson, D. Leech and D. Lebiwitz	Bob Swoger
Cassette LOAD/SAVE Problems Solved V	Don Lambert
Modem Time 1st MTERM II	Bob Swoger
Refining ZCOMM (MODEMing)	Joseph Rampolla
TS-2068 to Z-88 Transfer Via Modem	Dave Bennett
ZX-91 10 Years Later No.4	André Baune
VOLUME 2 NUMBER 3 Fall 1992	
Input/Output - A. Albrecht, D. Bennett, H. Polley, M. Binstock, J. Caldwell, B. Madaris, G. Parrish, and J. Rampolla	Bob Swoger
BYTE-BACK MD-68 Modem	Bob Swoger
Mineral Oil & The Printer Ribbon	Abed Kahale
Mineral Oil & The Edge Connector	Bob Swoger
Resetting Track 0 on Disk Drives	Don Lambert
Writing More Efficiently	Bill Harmer
ZX-91 10 Years Later No.5	André Baune
New Commands JLO SAFE V2.6	John Oliger
VOLUME 2 NUMBER 4 Winter 1992	
From The Editor's Desk - LogiCall 5.1	Bob Swoger
Input/Output - Marvin Johnson and Joseph Rampolla	Bob Swoger
50 TIPS (Prog. for the 2068 & Spectrum)	Don Lambert
VOLUME 3 NUMBER 1 Spring 1993	
Input/Output - D. Berry (Master Scribe), T. Simon and D. Lassov	Bob Swoger
Compound Interest and Investing	Don Lambert
Changing a REM statement in ZX-81 BASIC	Bill Harmer
Use Some Hitching Bits as a Flagman	Bill Harmer
VOLUME 3 NUMBER 2 Summer 1993	
SPDOS for RAMEX Mellina K (Disk drive)	Don Lambert
RecordKeeping	Abed Kahale
Input/Output - T. Graham, E. Radtke, W. Knaust. Keep'em Coming - G. Newkirk, D. Elliott, R. Madaris, F. Henn, A. Sweitzer, W Hanes, A. Syler, and L. Cottrell.	Editor
QZX Index	Alex Burr
Disk Life	Ted Jensen
Disk Life	Edward Snow
VOLUME 3 NUMBER 3 Fall 1993	
Spectrum AUTOSTART	Don Lambert
Re-Inking, Ribbon Resuscitation	John Wase
Keep'em Coming - J. Kuhlmann, D. Chattin, D. Stegman, L. Simon	Editor
Inexpensive Z-88 Parallel to Serial Converter	Jay Shepard III
Did You Know? (LarKen Tips)	Les Cottrell
Tasman 'B' Printer Interface (CPI)	Swoger / L. Kenny
Index Highlights - Articles	Editor
Public Domain Library Listings	Editor
VOLUME 3 NUMBER 4 Winter 1993	
Input/Output - Help - J. Payne, W. Horner, TV & Monitors, G. Parrish	Editor
Keep'em Coming - D. Lassov, F. Sklar, G. Simmons, Q. Kent.	
MEMOTECH MEMOPAK 64K	MEMOTECH
TS-2068 Talks to a PC by Modem	Abed Kahale
QL Hardware Project - Monitors	Bob Gilder
Tandy CM-11 Monitor	William Horner
MSDOS to LarKen & MSCRIPT	Les Cottrell
D.U.S. Disk Utility Software	Don Lambert
Turbo Switch for the ZX-81	Tony Willing
Using 16K MEMOTECH MEMOPAK	Don Lambert
Public Domain Library Update	Editor
VOLUME 4 NUMBER 1 Spring 1994	
Input/Output - D. Lassov, L. Howell, D. Williamson, G. Ray, D.Smith,	Editor
W. Knaust, R. Jelen, R. Shade, G. Parrish, R. Gowen, SUGWNY	Bob Swoger
Oliger Utilities	Don Lambert
QL Date Gate! - DBEASY -	Al Feng
LogiCall Review	Abed Kahale
TS-2068 - CMOS On Board	Richard Jelen
Ni-Cad Charger	Richard Jelen
TIMACHINE and the FDD	James Brezina
QL Video Output Circuit	Richard Jelen
QL Woes	Nazir Pashtoon

Moving Ramtop in ZX-81 TS-1000	Anthony Oresteen
ZX-81 POKEs and Calls	Anthony Oresteen
Frustrated PC Users Flood the Help Lines	Joan Kealy
D.U.S. Cheat Sheet	Abed Kahale
Why the Oliger Disk I/F with JLO SAFE is the Best for Your TS-2068	John Oliger
CW Decoder Circuit	Bob Swoger

VOLUME 4 NUMBER 2 Summer 1994

Input/Output - R. Jelen, E. Snow, R. Gilbert, G. Ray, R. Barnett,	
R. Hartung, R. Gowen, D. Lassov	Editor
LarKen Disk Interface Circuit	Les Cottrell
QLuMSi v.4.30	Al Feng
QLUSTer Upgrade	Al Feng
Relocating Machine Code - TS-2068	James Brezina
PARTS Inventory	Richard Jelen
Did You Try This?	David Lassov
SNUG Notice	Abed Kahale
T/SNUG QL Public Domain Library	Paul Holmgren

VOLUME 4 NUMBER 3 Fall 1994

Input/Output - W. Brzozowski, E. Phillips, A. Binstock, R. Shade,	Editor
Smith's Chart	Edwin Phillips
Computus Interruptus - 1	Wes Brzozowski
Batteries for the Z88 SNUG	Don Lambert
Adjust DATABASE Please	Al Feng
Information SuperHighway - Modems	Abed Kahale
Complex ASCII Rotation	Tim Swenson
Z-SI/O RS-232 Serial Interface	Abed Kahale
To PRINT or not to LPRINT	Jim Brezina

VOLUME 4 NUMBER 4 Winter 1994

Input/Output - B. Dyl, D. Lassov, D. Bennett,W. Brzozowski, T. Simon, R. Gowen, NESQLUG, SNUG, W. Mossberg	Editor
1994 UPDATE!	Frank Davis
A Surprise Box of Tricks - Z88	Hugh Howie
Z88 and its Power of Recovery	Hugh Howie
JLO with LKDOS Cartridge	Bob Swoger
SINCLAIR Rides the INTERNET	John Pazmino
Christmas Return Labels	Bob Swoger
Daisy Be Good - 1	David Lassov
Windows by Shade - 1	Robert Shade
Z88 Batteries	Mort Binstock
ZQA! 1991-94 Index to Articles	Editor

VOLUME 5 NUMBER 1 Spring 1995

Input/Output - Carlos Delhez, Robert Gilbert, Larry Crawford, Rod Gowen, Fred Henn, Greg Bridgewater, SNUG News, William Krossner.	Editor
Zebra FDD	Donald Lambert
Windows by Shade	Robert Shade
QL Hacker's Journal	Tim Swenson
QXL Notes	Al Feng
Z88 - My Memory Surprise	Hugh Howie
Z88 - Power to You	Hugh Howie
Computus Interruputs 2	Wes Brzozowski
TS Bulletin & BASIC N-L	William Harmer
Daisy Be Good II	David Lassov

VOLUME 5 NUMBER 2 Summer 1995

Input/Output - Les Cottrell, Tim Swenson, Robert Gilbert, Francine Sklar, Harry Miller, David Lassov, Joan Kealy, Bill Cable, Gilliam Parrish, Ferdinand Gunther, Carl Jones, David Lassov. Wire Guage	Editor
CAI/ESF Stringy Floppy	Donald Lambert
A Word to the Wise	Tim Swenson
Memories Mean a Lot	Abed Kahale
QXL Notes - Sequel	Al Feng
MDIR_BAS v1.05 MDIR_C	Al Feng
Lil'Amp Rides Again	Les Cottrell
Ql Corner	Bob Gilder
The Musical PC8300	Gilliam Parrish
QLuMSi -Recent Changes	Al Feng
Daisy Be Good III	David Lassov
Waging in the TSRoom	Donald Lambert
QL Hacker's Journal	Tim Swenson

VOLUME 5 NUMBER 3 Fall 1995

Input/Output - R. Hartung, P. Robinson, A. Albrecht, W. Harmer, D. Lambert, M. Binstock, F. Henn, M.van der Zwan, L. Moll, J. Dohany.	Editor
ZEUS Assembler	Alvin Albrecht
Towers of Hanoi	Alvin Albrecht
Digitizing & Sythesizing the 2068 Sound	Alvin Albrecht
TTSUC	Robert Swoger
Daisy Be Good IV	David Lassov
QL Hacker's Journal	Tim Swenson
Some Are QXL Notes	Al Feng
Windows by Shade 3	Robert Shade

VOLUME 5 NUMBER 4 Winter 1995

Input/Output - ZX-TEAM P. Liebert-Adelt, F. Henn, J. Moreno, D. Lassov, K. Posey, S. Dwyer, J. Clark, L. Cottrell, ZX81 LCD Screen.	Editor
Better Late than Never	Abed Kahale
PC Power Supply for your QL	Al Feng
QXL Totes	Al Feng
Daisy Be Good V	David Lassov
QHJ FreeWare	Tim Swenson
IBM Keyboard Interface for 1000/2068	Jack Dohany
50 Something	Abed Kahale

VOLUME 6 NUMBER 1 Spring 1996

Input/Output - Inter. ZX-81 Magazine, H. Scriven, P. Liebert-Adelt, G. Parrish, D. Lassov, F. Davis, J. Cruz-Figueroa, ZX-Team Magazine, QL Public Domain Library - S. Johnson, J. Moreno, TTSUC Library - G. Chamber, J. Dohany - IKI keyboard, L. Cottrell, F. Sklar, SCC BBS.	Editor
IKI Keyboard Interface (ZX-81, 2068)	Jack Dohany
ZX-81 Hi-Res?	T. Swenson- Internet
UDG's ZX-81	T. Swenson- Internet
DBEasy - 1.6 Review (QL)	Al Feng
Improving Sector_COPYing Program (QL)	Al Feng
Daisy Be Good - VI (2068)	David Lassov

VOLUME 6 NUMBER 2 Summer 1996

Input/Output - D. Johnson, K. Harbit, P. Liebert-Adelt, SJPD, GATOR, UPDATE!, J. Moreno SCC BBS, J. Kealy, K. Harbit, J. Shepard.	Editor
Expand the ZX-81 Memory to 32K	ZX-TEAM
DBEasy's EASY_OUT Custom (QL)	Al Feng
Windows by Shade - IV (2068)	Robert Shade
Errata - Sector_COPYing (QL)	Editor
QL Hacker's Journal (QL)	Tim Swenson
Telecommunication (2068)	David Lassov

Daisy Be Good - VII (2068)	David Lassov
The Web WWW	Abed Kahale

VOLUME 6 NUMBER 3 Fall 1996

Input/Output - D. Lassov, K. Watson, J. Kaczor, F. Davis, J. Pegram, A. Feng, J. Kealy, G. Parrish, GATOR, Internet virus, H. Raader, ZX-81 Hi-Res, J. Taylor, J. Moreno SCC BBS.	Editor
To Frank & Carol Davis	GATOR
Disk Doctor (2068)	George Chamber
QMOSAIC Chronicles (QL)	Al Feng
LogiCall Logic (2068)	GATOR
QL Hacker's Journal (QL)	Tim Swenson
Few Useful Z88 CLI Routines	Dave Bennett
AT Keyboard Interface ZX-81	Kai Fischer
Windows by Shade - V (2068)	Robert Shade
Daisy Be Good VIII (2068)	David Lassov
Surfing-the-Net with Sinclair	RAMTOP
TTSUC LarKen Disk Library	Editor

VOLUME 6 NUMBER 4 Winter 1996

Input/Output - RMG - R. Gowen, A. Tursucu, H. Howie, L. Cottrell, GATOR, T. Swenson, D. Lassov, S. Johnson, D. Lambert, J. Merz, P. Kwitkowski, A. Kahale, F. Davis.	Editor
Meet Phillip Kwitkowski	Bob Swoger
How to be Chic (2068/Specy)	Gil Parrish
QXL Ghosts (QL)	Al Feng
Daisy be Good - IX (2068)	David Lassov
Disk Drive Dress-Up	Les Cottrell
Telcommunication a la TS-2068	David Lassov
SeekQL 2.09 - Part 1 (QL)	Al Feng
ZX-81 Video Display System - Part 1	Wilt Rigter

VOLUME 7 NUMBER 1 Spring 1997

Input/Output - Did You Know? by Les Cottrell, QL Show	Editor
IRA Pay-Out	Don Lambert
LarKen Disk System	Gil Parrish
HJI Anounces New FreeWare	Tim Swenson
Aurora - The QL Graphic Card	
QL Hacker's Journal	Tim Swenson
Surfing the Net With the 2068	David Lassov
Parallel I/O Modification	Al Feng
QLUTter_BAS	Al Feng
Daisy Be Good X	David Lassov
ZX-81 Video Display II	Wilt Regter
TS-2068 Modem Compatible Serial I/F	Les Cottrell
SeekQL 2.09_Part 2	Al Feng

VOLUME 7 NUMBER 2 Summer 1997

Input/Output - WIDJUP CAD #3, 24-pin Printer Driver - Al Feng, TS-1500 - Gil Parrish, Battery Backup for the Z88 - Don Lambert.	Editor
QHJ - Off The Net	Tim Swenson
More Net Surf'n - 2068	Davisd Lassov
Insomnia Over the Z88	Don Lambert
Civilization ZX-81 Game	ZX-Team Magazine
Daisy Be Good XI	David Lassov
Running The QPC	Robert Hartung
Hack the ZX Spectrum	Les Cottrell

VOLUME 7 NUMBER 3 FALL 1997

Input/Output - Qpc, Ramtop Ts-2068 - Bob Swoger, Larken Graphic Adventure - Les Cottrell, DownLOAD from SOL BBS - Bob Swoger	Editor
Z88 Revisited	Don Lambert
D'You Know? Internet	Al Feng
More On QPC	Robert Hartung
QPC - The Missing Link	Al Feng
Internet to LarKen	Les Cottrell
Sequential Paragraph	David Lassov
Menu Magic	David Lassov
TeleCom Menu	David Lassov
ZEBRA on the Net	

VOLUME 7 NUMBER 4 Winter 1997

Input/Output - RMG Project, SmartWatch - David Lassov, E-Mail from Frank Davis, Recent QL FreeWare - Tim Swenson	Editor
More On QPC	Robert Hartung
QLATter	Al Feng
Sex for the ZX-81	Wilt Rigter
VAL Function	David Lassov
How to Hack on the Spectrum	Les Cottrell
Lunch Hour Activity	Les Cottrell

VOLUME 8 NUMBER 1 Spring 1998

Input/Output - RAM Storage - David Lassov, OS64 Manual, Garbage Gobbler David Lassov	Editor
RMG Project - Earl Kielglass, Robert Schimke, John Pegram, Gil Parrish, Hugh Scriven, Francine Sklar, Harry Miller, Bob Barnett, Thoresen, Dorinson, Joan Kealy.	Donating Members
QL Hacker's Journal	Tim Swenson

VOLUME 7 NUMBER 1 Spring 1997

Input/Output - Did You Know? by Les Cottrell, QL Show	Editor
IRA Pay-Out	Don Lambert
LarKen Disk System	Gil Parrish
HJI Anounces New FreeWare	Tim Swenson
Aurora - The QL Graphic Card	
QL Hacker's Journal	Tim Swenson
Surfing the Net With the 2068	David Lassov
Parallel I/O Modification	Al Feng
QLUTter_BAS	Al Feng
Daisy Be Good X	David Lassov
ZX-81 Video Display II	Wilt Regter
TS-2068 Modem Compatible Serial I/F	Les Cottrell
SeekQL 2.09_Part 2	Al Feng

VOLUME 7 NUMBER 2 Summer 1997

Input/Output - WIDJUP CAD #3, 24-pin Printer Driver - Al Feng, TS-1500 - Gil Parrish, Battery Backup for the Z88 - Don Lambert.	Editor
QHJ - Off The Net	Tim Swenson
More Net Surf'n - 2068	Davisd Lassov
Insomnia Over the Z88	Don Lambert
Civilization ZX-81 Game	ZX-Team Magazine
Daisy Be Good XI	David Lassov
Running The QPC	Robert Hartung
Hack the ZX Spectrum	Les Cottrell

VOLUME 7 NUMBER 3 Fall 1997

Input/Output - Qpc, Ramtop Ts-2068 - Bob Swoger, Larken Graphic Adventure - Les Cottrell, DownLOAD from SOL BBS - Bob Swoger	Editor
Z88 Revisited	Don Lambert
D'You Know? Internet	Al Feng
More On QPC	Robert Hartung
QPC - The Missing Link	Al Feng

Internet to LarKen	Les Cottrell
Sequential Paragraph	David Lassov
Menu Magic	David Lassov
TeleCom Menu	David Lassov
ZEBRA on the Net	

VOLUME 7 NUMBER 4 Winter 1997

Input/Output - RMG Project, SmartWatch - David Lassov, E-Mail from Frank Davis, Recent QL FreeWare - Tim Swenson	Editor
More On QPC	Robert Hartung
QLATter	Al Feng
Sex for the ZX-81	Wilt Rigter
VAL Function	David Lassov
How to Hack on the Spectrum	Les Cottrell
Lunch Hour Activity	Les Cottrell

VOLUME 8 NUMBER 1 Spring 1998

Input/Output - RAM Storage - David Lassov, OS64 Manual, Garbage Gobbler - David Lassov, Al Feng, Jack Boatwright, Fred Henn, Dave Bennett, Rod Gowen, J Shepard, GATOR, Bill Cable, George Chambers, Dave Lebowitz, Doug Gellispie, Ken Harbit, Andrew Dansby, Peter Liebert.	Editor
RMG Project - Earl Kielglass, Robert Schimke, John Pegram, Gil Parrish, Hugh Scriven, Francine Sklar, Harry Miller, Robert Barnett, Thoresen, Dorinson, Joan Kealy.	Donating Members
QL Hacker's Journal - 27	Tim Swenson

VOLUME 8 NUMBER 2 SUMMER 1998

Input/Output – RMG, Jack Boatwright, Joan Kealy, Jochen Merz, Alvin Albrecht, Andrew Dansby, Rod Gowen, Vacuum Tubes, Russ Perry, Jay Shepard, GATOR, Termax, Bill McKelvey, George Chambers, Frank Davis, Tim Swenson, Al Feng, David Lassov, Rod Gowen	Editor
Timex-Sinclair inventory	Jack Boatwright
QLay .85	Al Feng
QPC Update	Robert Hartung
Review	Gil Parrish
Surf'n the Net	David Lassov
How to Hack the ZX Spectrum -3	Les Cottrell
MaxCom	David Lassov
RMG List	Rod Gowen
Internet Glossary	Editor
ZQA! Index 1997-1998	

VOLUME 8 NUMBER 3 FALL 1998

Input?Output – QPC – Hartung, LarKen Boards – GATOR, Rod Gowen, David Lassov, Jay Shepard, Louis Florit, Josh Payne, Jack Boatwright, HP 500C Printer fire, Puzzle – Gil Parrish.	Editor
QL Hacker's Journal	Tim Swenson
ZipaDee_BAS	Al Feng
QLABer Update	Al Feng
QLATer	Al Feng
How to Hack on the ZX Spectrum – 4	Les Cottrell
RMG Inventory	Jack Boatwright

VOMUME 8 NUMBER 4 Winter 1998

Input/Output – Puzzle solved – Peter Liebert-Adelt, GATOR, Les Cottrell, Jack Boatwright, David Solly, Inventory List – Jay Shepard, Louis Florit, Frank Davis, Sinclair Show, John Pegram, Gerald Anson, West Coast Sinclair Show – Tim Swenson, David Lassov, GOODBYE Sinclair – Frank Davis	Editor
FOP_OVER	Al Feng
LarKen/PASCAL	David Solly
LarKen/HiSoft	David Solly
Speed Comparison	David Solly
QL Hacker's Journal – 29	Tim Swenson

VOMUME 9 NUMBER 1 Spring 1999

Input/Output – QL-East Coast, NESQLUG – Ed Kingsley, Jose Moreno, David Solly, Peter Liebert-Adelt, Clive Reveals ZX-2000 – Jack Boatwright, Jeff Burrell, Don Oviatt, Earl Kielglass, Bill Mckelvey, Bill Marriott, Frank Davis, Memotech – Peter, Luke Perry, Conrad Perfett, Keith Watson, Mark McCann, SOLD – Frank Davis, Timex – Sally Jagello, Bob Hartung – QL News, Fred Henn, Home Electronics Service – John Rish.	Editor
TS-2068 Talks to a PC by Modem	Abed Kahale
QL Hacker's Journal #30	Timothy Swenson
CHR$ Emulation – HiSoft Pascal	David Solly
How to Hack on the Spectrum	Les Cottrell

VOLUME 9 NUMBER 2 Spring 1999

Input/Output – David Solly (Larry Kenny), Glen Goodwin, Jack Boatwright Boatwright (Sir Clive Sinclair), Fred Henn (CTM), Don Oviatt, Rod Gowen.	Editor
QL Today Notes	Robert Hartung
Font Loader – HiSoft Pascal	David Solly
West Coast Sinclair Show	Tim Swensen
Jet Printer Ink	Editor
TS-2068 Internal ROM By-Pass	William Pedersen
How to Hack on the Spectrum VI	Les Cottrell

VOLUME 9 NUMBER 3 Fall 1999

Input/Output – Joan Kealy, Dane Stegman, Stephen Waldman, Richard Burt, William McBrine – TS-2068 Users 2068@unixville .com, Joe Rampolla,	Editor
QL Hacker's Journal	Tim Swensen
TS-2068 ROM ByPass	William Pedersen
Function for UDGs HiSoft	David Solly
How to Hack on the Spectrum VI	Les Cottrell

VOLUME 9 NUMBER 4 Winter 1999

Input/Output – Joe Rampolla, Antonio Castro, Jack Boatwright Boatwright, Jim Mack, William Girnius, Larry Kenny, Rod Gowen, Al Feng, Mike Kondrac, Sinclair Z1 Radio, Visit to Larry Kenny's Page, The Train, Douglas Dunbar, Dean Mikolajczyk, Alvin Albrecht, J. Shepard, Joan Kealy, Robert Gilbert, David Lassov, Donald Lambert, Luke Perry, Johon Kasza, Frank Davis, Jeff Burrell, Dane Stegman.	Editor
Changing Scren Attrb. HiSoft	David Solly
Z80 WorkShop	E. A. Parr

VOLUME 10 NUMBER 1 Spring 2000

Input/Output –Paul Anderson, Jack Boatwright, Dane Stegman, Joan Kealy, Andy Barber, Bill Mathis, Don Lambert, David Lassov, Fred Henn, Jeff Burrell, Joe Rampolla, Earl Kielglass, Harry Miller, Luke Perry, John Oliger,	Editor
Fun With Hardware I	Jeff Burrell

QL Hacker's Journal	Tim Swenson
Diskette Menu Program	David Solly
ZX MicroDrive	David Solly
TS-1000 I/O Board	Joe Rampola

VOLUME 10 NUMBER 2 Summer 2000

Input/Output – Joe Rampolla, Neil, John Pegram, Al Feng, Davis Leech, Jack Boatwright, Jeff Burrell, Rod Gowen, Dane Stegman, Robert Webster, Tom Parks, Robert Swoger, Hugh Scriven, Fred Henn, Bill Cable, Keith Watson, Don Lambert, Jay Shepard, Juan Caprio, CATUG Members, QL Users E-Mail (Quanta)	Editor
7-Bit ASCII Filter for LarKen Disk I/O	David Solly
How to Hack on the Spectrum	Les Cottrell
Memory Improves with Age	William Pederson

VOLUME 10 NUMBER 3 Fall 2000

Input/Output – Luke perry, Rod Gowen, Darrell Hamilton, Frank Mills, David Lassov, Leon Howell, Dave Bennett, Wilf Rigter, David Solly	Editor
ZX-81 Video Display System	Wilf Rigter
Z80 LCD	Wilf Rigter

VOLUME 10 NUMBER 4 Winter 2000

Input/output --- Don Lambert,Peter Liebert-Adelt, Robert Gilbert, Jeff Burrell, Joe Rampolla, Al Feng, Robert Swoger, Louis Florit, David Solly, Keith Watson, Jonny Red, Alvin Albrecht, John Donaldson, Luke Perry,	Editor
QL Hacher's Journal	Tim Swenson
Fourth	Glen Haydon
TS – ZX Spectrum	David Solly
EZ80 Micro Computer	Wilf Rigter
Z88	Dave Bennett

VOLUME 11 NUMBER 1 Spring 2001

Input/Output --- Paul Anderson, John Gaylor, Stephen Beals, Peter Liebert, William McBrine, Luke Perry, Louis Florit, Johnny Red, J. Shepard, Tim Swenson, John Donaldson, Robert Swoger, Dane Stegman, David Solly, Joan Kealy. JLO PD	Editor
Setting Caps Lock	David Solly
Graphics HiSoft	David Solly
Flasht Listings	David Solly

VOLUME 11 NUMBER 2 Summer 2001

Input/Output --- David Solly, Tim Swenson, J. Shepard, Don Lambert, Dean Mikolajczk, Jack Boatwright, Louis Florit, Peter Liebert, Paul Harris, Luke Perry. JLO PD.	Editor
MSCRIPT Command Summary	David Solly
The Rings of Saturn	David Solly
Our Past is Dying	David Solly
LarKen Tape Backup.	George Chambers

VOLUME 11 NUMBER 3 Fall 2001

Input/Output— Will Knight, Joan Kealy, Tim Swenson, John Donaldson, Leon Howell, Don Lambert, J. Shepard, Jose Moreno, Paul Harris, Peter Liebert, Jeff Burrell, Dave Bennett, Richard Tryon, Uwe Schonewolf, ZX-96, Union City Man.	Editor
Structured SuperBasic	
Trials and Tribulations	Don Lambert
True Type Fonts	David Solly
Spectrum Fonts	Darko Stanicic
Creating TS-2068 Fonts	
Emulators	Warajavo
Z88 Source Book	

VOLUME 11 NUMBER 4 Winter 2001

Input/output — Dean Mikolajczk, Peter Liebert, Robert Webster, David Solly, MSCRIPT, TS-2068 Disk Drive, David Hartman, Dane Stegman, Burke Mawby.	Editor
Another Monitor Adapter	Les Cottrell
Making Cassette Tapes	Don Lambert
Z88 Source Book - 2	

VOLUME 12 NUMBER 1 Spring 2002

Input/output — Luke perry, Mike Ingall, Earl Kielglass, Paul Holmgren, QL Show	Editor
Z88 Source Book – 3	
QL Hacker's Journal	Tim Swenson
The TS-2068 SCLD	Victor Acuna

VOLUME 12 NUMBER2 Summer 2002

Input/Output — Ruth Fegley, Jack Boatwright, Ed Kingsley, Al Feng, Bob Hartung, Don Lambert, Louis Florit, Bob Swoger, Tony Firshman.	Editor
QPC2 for Windows	Robert Hartung
Tasword to Mscript	David Solly
Sinclair ASCII to IBM	David Solly
Internet File Format On The QL	Timothy Swenson
Z-88 Source Book sec. 4	

VOLUME 12 NUMBER 3 AUTUMN 2002

Input/Output --- Hector Picone, Don Lambert, Bjorn, James Cole, Scott Rossell, Joan Kealy, Tony Firshman, Ruth Fegley, David Solly.	Editor
SuperBASIC	Al Boehm
LarKen Disk Drive Manual	Larry Kenny

VOLUME 12 NUMBER 4 WINTER 2003

Input/Output — Jack Boatwright, Loic Daneels, David Solly, Luke Perry, Louis Florit, William McBrine, Les Cottrell, Bob Hartung, Jim Ditton, Keith Watson, Don Lanbert, Bob Swoger, Callum Davidson, Edwin Kranpitz, Jonny Red,	Editor
TS-2068 Cassette Meter	Don Lambert
Z-88 Source Book Conclusion	

12th Anniversary NUMBER 4 Spring 2003

ZQA! Signing Off	Robert Swoger Abed Kahale
1991 ZQA!	Robert Swoger
From the Chairman's Desk	Donald Lambert
Input/Output — Al Boehm, Les Cottrell, Keith Watson, Bob Swoger, John Simpson, Peter-Liebert-Adelt, Glen Goodwin, Mark Anderson, Andre Baune, David Solly, Joan Kealy, Callum Davidson, Tony Firshman, Josef Rampolla.	Editor
Can NASQLUG Take Up The Slack	Al Boehm
NESQLUG	Al Boehm
Timex Emulation	Callum Davidson
C Programming	Alvin Albrecht
A Fast Pattern Flood Fill	Alvin Albrecht
Members Addresses	
ZQA! Index	

Back copies mailed for $250 each: -

A. KAHALE
432 WEST OAKS TRL
WOODSTOCK GA 30188-7358